VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Metody detekce komunit ve vícevrstvých sítích

Community Detection Methods in Multi-layer Networks

2020 Matej Kubinec

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Matej Kubinec**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2612T025 Informatika a výpočetní technika

Téma: Metody detekce komunit ve vícevrstvých sítích
Community Detection Methods in Multi-layer Networks

Jazyk vypracování: čeština

Zásady pro vypracování:

Cílem práce je implementace vybraných metod detekce komunit v prostředí vícevrstvých sítí. Preferován je jazyk C#.

1. Rešerše obdobných řešení.
2. Implementace vybraných metod detekce komunit v prostředí vícevrstvých sítí.
3. Implementace webové aplikace využívající implementované metody.
4. Dokumentace s využitím standardů softwarového inženýrství.

Seznam doporučené odborné literatury:

[1] Dickison, M. E., Magnani, M., & Rossi, L. (2016). Multilayer social networks. Cambridge University Press.
[2] Bianconi, G. (2018). Multilayer Networks: Structure and Function. Oxford university press.

Dále podle pokynů vedoucího práce.

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Mgr. Miloš Kudělka, Ph.D.**

Datum zadání: 01.09.2019
Datum odevzdání: 30.04.2020

doc. Ing. Jan Platoš, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prehlasujem, že som túto diplomovú prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave .....

Poďakovanie

Na tomto mieste by som sa rád poďakoval doc. Mgr. Milošovi Kudělkovi, Ph.D. za vedenie tejto práce. Ďalej by som sa chcel poďakovať mojej rodine a snúbenici za podporu.

Abstrakt

Táto práca sa zaoberá problematikou detekcie komunít na viacvrstvových sieťach. Cieľom tejto práce bola implementácia vybraných metód na detekciu komunít a ich následné použitie vo webovej aplikácie. Výsledkom práce je webová aplikácia umožňujúca analyzovať a vyhodnocovať viacvrstvové siete z pohľadu detekcie komunít.

Kľúčové slová: viacvrstvové siete, detekcia komunít, MNCD, splošťovanie, C#, React

Abstract

This thesis deals with the topic of community detection in multi-layered networks. The goal of this work was to implement selected community detection methods and its subsequent use in a web application. The result of this thesis is a web application, which enables to analyze and evaluate multi-layer networks from the perspective of communities.

Keywords: multi-layer networks, community detection, MNCD, flattening, C#, React

Obsah

Zoznam použitých skratiek a symbolov

SNA	–	Social Network Analysis
CLECC	–	Cross-Layer Edge Clustering Coefficient
MNCD	–	Multi-layer Network - Community Detection
API	–	Application Programming Interface
HTTP	–	Hypertext Transfer Protocol
JSON	–	JavaScript Object Notation
SVG	–	Scalable Vector Graphics
PNG	–	Portable Network Graphics
SPA	–	Single Page Application

Zoznam obrázkov

Zoznam tabuliek

Zoznam výpisov zdrojového kódu

1 Úvod

Komplexné systémy sú často charakterizované množstvom interakcií medzi ich jednotkami. Tieto interakcie je výhodné skúmať aj z pohľadu sietí, ktoré ich umožňujú zakódovať do svojej štruktúry. Vznikajú nám siete obsahujúce množstvo rôznych typov hrán reprezentujúcich iné typy spojenia. Viacvrstvové siete nám ponúkajú iný typ pohľadu na takéto siete.

Aj keď matematická formulácia viacvrstvových sietí je pomerne mladá, koncepty, ktoré obsahuje, sú používané už od 60-tych rokov 20. storočia. V tomto období sa rozvíjala analýza sociálnych sietí a medzi prvé prístupy použitia viacvrstvových sietí môžeme považovať kvantitatívny prístup k analýze viacvrstvových sociálnych sietí autora White a iných [1]. Spomenutí autori navrhli jednoduchú trojvrstvovú štruktúru, kde uzly reprezentujú vedcov oboru biomedicíny. Jednotlivé vrstvy reprezentujú obojsmerné osobné prepojenie, známosť z pohľadu jedného vedca a prepojenia na poslednej vrstve udávajú stav, kde ani jeden z vedcov navzájom nepozná druhého. S využitím takejto siete, navrhli metódu ako odhaliť súvislé bloky na viacerých vrstvách, pričom túto metódu môžeme pokladať za jeden z prvých algoritmov na detekciu komunít.

Ako moju diplomovú prácu som si vybral práve metódy detekcie komunít na viacvrstvových sieťach. Zaujal ma nový pohľad na riešenia problémov, ktoré som doposiaľ riešil len v priestore sietí s jednou vrstvou a jedným typom hrany. Analýza z pohľadu viacvrstvových sietí prináša množstvo nových problémov, keďže siete už nie sú triviálne a neraz nie je jasné, ako k niektorým problémom pristúpiť. Jeden z najčastejších problémov v oblasti analýzy sietí je detekcia komunít, ktorá už pri jednovrstvových sieťach nie je vždy dobre definovaná a samotná definícia pojmu komunita, nie je totožná naprieč rôznymi prístupmi a inak tomu nie je ani v prostredí viacvrstvových sietí. Detekciu komunít považujem za jeden z významných prístupov pri analýze dát, často môžeme nájsť v dátach, ale aj všade okolo nás, isté formy prvkov, ktoré sú si blízke. Stačí sa pozrieť na fenomén sociálnych sietí, kde môžeme vidieť, že ľudia často vytvárajú veľmi prepojené skupiny a to nielen z dôvodu, že sú si blízky, ale aj z pohľadu spoločných záujmov, záľub a ďalších spoločných prepojení. Keď sa zameriame bližšie na tieto prepojenia, môžeme v nich vidieť, že hoci sú si ľudia voči sebe neznámi, cez spoločných známych sú si blízky, a na základe fenoménu malého sveta, nám často stačí približne šesť prepojení aby sme spojili dvoch ľubovoľných ľudí prostredníctvom známych [2]. Tento fenomén je často reprezentovaný jednoduchou sieťou, a môže byť zaujímavé pozrieť sa na tento fenomén aj z pohľadu vrstiev, kde každá vrstva môže predstavovať istú oblasť záujmu, známosť alebo ľubovoľné iné prepojenie.

V mojej práci vám najskôr bližšie priblížim svet viacvrstvových sietí, zadefinujem pojmy, ukážem príklady takýchto sietí vo svete a ako sa dajú reprezentovať. V nasledujúcej kapitole priblížim prístupy akými sa dá upraviť viacvrstvová sieť na jednovrstvovú. Ďalej vám priblížim prehľad riešení problému detekcie komunít, či už na jednovrstvových sieťach, ako aj na sieťach viacvrstvových. V posledných častiach sa budem venovať praktickému riešeniu týchto problémov pomocou knižnice a webovej aplikácie, ktoré som v rámci praktickej časti implementoval.

2 Rešerše

Chuan Wen, Loe a Henrik Jeldtof Jensen sa zaoberajú v Comparison of Communities Detection Algorithms for Multiplex [3] porovnávaním prístupov detekcie komunít na viacvrstvových sieťach, okrem teoretických základov porovnávajú výsledky rôznych algoritmov na dátach.

Santo Fortunato sa venoval v diele Community detection in graphs [4] detekcii komunít na jednovrstvových sieťach, poznatky z tohto diela som využil v sekcii 6.2.

V knihe Multilayer Social Networks sa Mark E. Dickison, Matteo Magnani a Luca Rossi [5] zaoberajú viacerými odvetviami viacvrstvových sietí, vedomosti o zjednodušovaní sietí a splošťovaní som využil v kapitole 5.2 a obecné znalosti o viacvrstvových sieťach v kapitole 4.

ABACUS: frequent pAttern mining-BAsed Community discovery in mUltidimensional networkS od Michele Verlingerio, Fabio Pinelli a Francesco Calabrese [6] som využil pri popise dolovania frekventovaných položkových množín 6.4 a pri implementácií algoritmu ABACUS.

Dielo od Piotr Bródka, Tomasz Filipowski1, Przemysław Kazienko An Introduction to Community Detection in Multi-layered Social Network [7] definuje mieru CLECC pre hrany a algoritmus detekcie komunít založený na tejto miere. Poznatky som použil pri implementácií daného algoritmu, ako aj v sekcii o detekcii komunít pomocou detekcie mostov.

Fast unfolding of communities in large networks od Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte a Etienne Lefebvre [8] prichádzajú s metódou Louvain, ktorá je rýchlou metódou pre detekciu komunít s využitím optimalizácie modularity. Z diela som vychádzal pri implementácii algoritmu ako aj v sekcii o optimalizácie modularity.

3 Dátové sady

V tejto sekcii priblížim dátové sady viacerých viacvrstvových sietí, s ktorými budem v ďalších kapitolách pracovať.

Florentine Dáta boli zozbierané Johnom Padgettom z historických dokumentov popisujúcich vzťahy medzi 16 politicky prominentnými rodinami vo Florencii okolo roku 1430. Obsahuje dve vrstvy, vrstvu obchodným vzťahov, určenú na základe finančných prepojení ako sú pôžičky, úvery a spoločných obchodných partnerstiev a vrstvu manželstva. V dátach figurujú dve silné rodiny, Medici a Strozzi, čím sú dáta vhodné na testovanie metód detekcie komunít. Dáta obsahujú ako som už spomínal dve vrstvy, 16 aktérov a 25 hrán. Dostupné sú z `http://multilayer.it.uu.se/datasets.html`.

Monastery Skupina mníchov, bola oslovená, aby každý z nich špecifikoval troch mníchov a priradil každého z nich ku jednej zo štyroch párov pozitívnych/negatívnych vzťahov. Týmito pármi sú úcta/neúcta, náklonnosť/odpor, pozitívny vplyv/negatívny vplyv a chvála/vina. Limit troch preferencií môže ovplyvňovať merania na základe stupňov, keďže výstupný stupeň každého z aktérov je tri. Dáta obsahujú 10 vrstiev, 18 aktérov a 510 hrán, dostupné sú z `http://multilayer.it.uu.se/datasets.html`.

Bankwiring Dáta, prezentované Roethlisberger a Dickson v roku 1939, popisujú 14 zamestnancov Hawthrone elektrárne pracujúcich v bank wiring-u. Zamestnanci majú rôzne role (dvaja inšpektori, traja spájkovači, a deväť elektrikárov), čím sú dáta užitočné pre testovanie rolí/metod detekcie pozícií. Jednotlivé vrstvy popisujú nápomocnosť v práci, argumentáciu o otvorených oknách, priateľstvo, negatívne správanie voči sebe, počet výmen zmeny. Dáta obsahujú šesť vrstiev, 14 aktérov, 110 hrán a sú dostupné z `http://multilayer.it.uu.se/datasets.html`.

Tailorshop Zozbierané Kapferom v roku 1972 reprezentujú pracovné a priateľské interakcie medzi 39 pracovníkmi v krajčírstve. Sú dostupné dve verzie sociálnej siete, zozbierané boli v rozličnom čase. Dáta obsahujú štyri vrstvy, 39 aktérov a 552 hrán. Dostupné sú z `http://vlado.fmf.uni-lj.si/pub/networks/data/UciNet/UciData.htm#kaptail`.

AUCS Tieto anonymizované dáta popísané Rossim a Magnanim (2015), boli zozbierané na výskumnej katedre univerzity a obsahujú päť online a offline vrstiev. Populácia sa skladá zo 61 zamestnancov (z celkového počtu 142), ktorý sa rozhodli zapojiť do štúdie. Zahrnutí sú profesori, študenti na postgraduálnom štúdiu aj administratívny pracovníci. Dáta obsahujú päť vrstiev, 61 aktérov a 620 hrán, dostupné sú z `http://multilayer.it.uu.se/datasets.html`.

4 Viacvrstvové siete

4.1 Základné pojmy

Pojmy, ktoré sa používajú pri viacvrstvových sieťach pochádzajú z rôznych odvetví, ktoré sa vyvíjali navzájom, od sociálnych vied až po informatiku. Terminológia sa vďaka tomu líši vzhľadom na oblasť v ktorej sa pohybujeme, ale najzaužívanejšia forma pochádza z oblasti analýzy sociálnych sietí, grafov a tou sa budem aj v mojej práci riadiť.

Pred tým, než si zadefinujeme viacvrstvovú sieť, si musíme zadefinovať sieť s jednou vrstvou. Sieť sa matematicky najčastejšie definuje ako graf.

Definícia 1 (Graf) *je dvojica $G = (V, E)$, kde V je množina uzlov (vrcholov) a E je množina hrán $E \times V\ V$.*

Grafy môžeme na základe orientácie hrán rozdeliť na orientované a neorientované. Hrany môžu mať rôzne atribúty, pričom najčastejšie je používané číslo, ktoré udáva váhu hrany vzhľadom k ostatným. Grafy, ktoré obsahujú takéto hrany nazývame ohodnotené.

Na rozdiel od jednoduchých grafov nám pribudla množina vrstiev siete L. V obecnej forme môže existovať hrana nielen medzi uzlami v rámci jednej vrstvy, nazývané *vnútro-vrstvové*, ale môžu existovať aj hrany medzi uzlami z rozličných vrstiev, nazývané *medzi-vrstvové*. Vzhľadom na pôvod pojmov z oblasti SNA, sa nepoužíva pojem uzol, ale pojem aktér (ang. Actor). Tento rozdiel vyplýva zo skutočnosti, že pojem uzol je matematický pojem, kdežto pojem aktér, vyjadruje entitu reálneho sveta a je vhodnejší na použitie v oblasti SNA. V terminológii viacvrstvových sietí chápeme pojem aktér ako osobu alebo organizáciu, ktorá môže mať vzťah s ostatnými aktérmi, kdežto uzlom sa chápe špecifický aktér na určitej vrstve. Na základe týchto informácií definujeme viacvrstvové siete nasledovne.

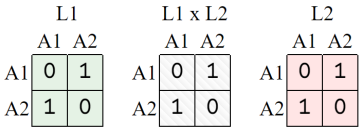
Definícia 2 (Viacvrstvová sieť) *je štvorica $M = (A, L, V, E)$, kde (V, E) je graf a $V \subset A \times L$, A je množina aktérov a L je množina vrstiev.*

Jednotlivé vrstvy reprezentujú určitú vlastnosť alebo aspekt, ktorý charakterizuje aktérov v danej vrstve. Týmito vlastnosťami môžeme chápať napríklad určitý typ vzťahu na sociálnej sieti, rozdielny typ hrany ako aj zachytenie časovej zložky, kde každá vrstva reprezentuje stav siete v určitom čase.

4.2 Reprezentácia

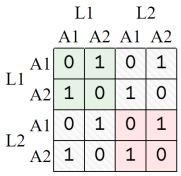
Viacvrstvové siete sa dajú reprezentovať rôznymi spôsobmi a na základe potrebných vlastností sú niektoré reprezentácie vhodnejšie ako iné, v tejto podkapitole vám niektoré z nich priblížim.

Zoznam hrán je jeden z najjednoduchších reprezentácií viacvrstvovej siete. Pri každej hrane je nutné zachytiť aktérov, ktorých vzťah hrana vyjadruje, ako aj vrstvy, v ktorých sa aktéri nachádzajú. Okrem toho hrana môže obsahovať ďalšie údaje, ako sú orientácia hrany, váha hrany a iné atribúty. Výhodou tejto reprezentácie je ľahká implementácia, možnosť zachytiť rôzne druhy atribútov pri každej hrane. Hlavnou nevýhodou je nemožnosť zachytiť aktérov, ktorý nie sú prepojený žiadnou hranou. Ak je nutné takýchto aktérov spracovávať, potrebujeme ďalší zoznam. Ďalšou nevýhodou sú niektoré algoritmy, ktoré predpokladajú inú reprezentáciu na to aby boli efektívne. Pri implementácií algoritmov a návrhu aplikácie som používal práve túto reprezentáciu rozšírenú o zoznam aktérov.



Obr. 1: Matica susednosti

Množina matíc susednosti je ďalšou formou reprezentácie, kde každá z matíc predstavuje jednu z vrstiev. Stĺpce a riadky reprezentujú aktérov, a jednotlivé hodnoty určujú váhu danej hrany. Pri nulovej váhe predpokladáme, že daná hrana neexistuje. Výhodou takejto reprezentácie je hlavne ľahký prístup k určitej hrane. Nevýhodami sú väčšia pamäťová náročnosť v prípade riedkych grafov a nemožnosť zachytiť hrany medzi jednotlivými vrstvami bez toho, aby sme výrazne nezvyšovali pamäťové nároky tým, že by sme ukladali aj matice susednosti medzi jednotlivými vrstvami.



Obr. 2: Supra-matica susednosti

Supra-matica susednosti sa dá chápať ako iný zápis množiny matíc susednosti, kde jedna matica reprezentuje všetky matice susednosti. Vzniká nám matica o rozmeroch $N * L \times N * L$, kde N je počet aktérov a L je počet vrstiev, čo v prípade veľkých a hlavne riedkych matíc predstavuje veľkú pamäťovú záťaž. Tieto matice sú reprezentáciou unikátnou reprezentáciou tenzorov štvrtého rádu. Supra-matice susednosti majú štvorcové rozmery a sú vhodnou reprezentáciou v prípade výpočtov, ako sú výpočet vlastných vektorov [9] alebo náhodná prechádzka [10]. Reprezentáciu som v projekte nepoužil.

4.3 Aplikácie viacvrstvových sietí

Viacvrstvové siete nachádzajú uplatnenie vo viacerých sférach, ponúkajú komplexnejší pohľad oproti klasickým sieťam, ktoré často zachycujú len zjednodušenú formu daného systému. Príklad takého zjednodušenia môžeme vidieť na analýze dát zo sociálnych sietí, kde nám vystupujú rovnakí ľudia, no sú prepojený rôznymi druhmi vzťahov, či už sa jedná o prepojenie na inej sociálnej sieti, alebo iná relácia v tej istej sieti. V týchto prípadoch nám viacvrstvové siete ponúkajú praktickejší prístup k zachytení takýchto vzťahov. V tejto kapitole postupne ponúknem náhľad na oblasti, kde nachádzajú viacvrstvové siete najväčšie využitie.

Sociológia Viacvrstvové siete nachádzajú široké uplatnenie v oblasti sociológie a vzťahov. Ako som uvádzal už v úvode, v sociológii majú pôvod samotné viacvrstvové siete, a už v období 80. rokov 20. storočia sa využívali na modelovanie vzťahov. V súčasnosti sa využívajú vo veľkej miere na modelovanie online interakcií medzi užívateľmi, ako je tomu napríklad pri analýze Pardusovej sociálnej hry Szellom a inými [11]. V dátach tejto online hry pre viacerých hráčov identifikovali šesťvrstvovú štruktúru, ktorá popisuje vzťahy medzi hráčmi, ich komunikáciu, aktivity a iné spojenia.

Technické systémy Jednou z aplikácií viacvrstvových sietí v technológii je ich využitie pri štúdiu prepojených systémov, teda systémov, kde správna funkčnosť jednotlivých systémov závisí vo veľkej miere na funkčnosti ostatných. Jednou z takýchto sietí sú aj siete na prenos elektrickej energie, ktorých funkčnosť často závisí aj na iných sieťach, napríklad komunikačných. Výpadkom takejto siete z 28. septembra 2003 v Taliansku sa zaoberali Buldyrev a iný [12], kde ako model použili siete, medzi ktorými určité uzly na sebe záviseli. Ďalším príkladom technických systémov sú transportné siete, kde sú viacvrstvové siete dobrou reprezentáciou komunikácií, kde rozličné druhy komunikácie predstavujú odlišné vrstvy. Takúto reprezentáciu použili Halu a ostatný [13] pri modelovaní Indickej leteckej a železničnej dopravy.

Biomedicína Použitie metodológie komplexných sietí spôsobilo v oblasti biomedicíny menšiu revolúciu, umožnilo pristupovať k systémovej medicíne, kde sa neštudovali len elementy z ktorých sa živé bytosti skladajú, ale aj sa systematicky analyzovala ich vzájomná komunikácia a interakcia. Dve hlavne oblasti, kde sa takýto prístup osvedčil, sú charakterizácia zloženia buniek, špeciálne génov [14], proteínov a metabolitov, druhou oblasťou je oblast neurológie, kde sa pracuje so sieťami neurónov anatomicky aj funkčne [15].

Okrem týchto oblastí sa viacvrstvové siete využívajú aj v ekológii (interakcia medzi paviánmi [16]), pri štúdiu klímy (viacvrstvová sieť reprezentujúca atmosférické vrstvy [17]), ekonómie (analýza medzinárodnej obchodnej siete a dopade výpadkov [18]) a iných odvetviach.

5 Prevod na jednovrstvovú sieť

Pred tým ako sa pozrieme na samotnú detekciou komunít, si priblížime možnosti, akým spôsobom prevádzať viacvrstvovú sieť na jednovrstvovú. Tento prevod je dôležitý aj z pohľadu detekcie komunít tým, že nám umožňuje aplikovať algoritmy, ktoré boli navrhnuté len pre jednovrstvové siete na viacvrstvových sieťach. Existuje viacero prístupov ako takýto prevod vykonať a v tejto kapitole vám ich priblížim.

Ako prvé si priblížime dva prístupy redukcie siete a to globálne zjednodušenie a lokálne zjednodušenie.

5.0.1 Globálne zjednodušenie

Prístup globálneho zjednodušovania je vhodný pre automatickú voľbu množiny vrstiev tak, aby sme stratili, čo najmenej informácií. Problém globálneho zjednodušenia bol definovaný De Domenicom a ostatnými [19]. Na proces globálneho zjednodušovania potrebujeme dve miery, prvou z nich je spôsob ako vypočítať podobnosť jednotlivých vrstiev a druhou je miera, ktorá vyjadruje kvalitu spojenia vrstiev. Proces následne pokračuje v štyroch krokoch.

1. Pre každý pár vrstiev je vypočítaná ich podobnosť (alebo vzdialenosť)
2. Vykoná sa hierarchické zhlukovanie nad vrstvami, vnútorné uzly reprezentujú spojenie vrstiev
3. Na každej úrovni dendrogramu sa vrstvy s najväčšou podobnosťou spoja a ich spojenie je vyhodnotené kvalitatívnou funkciou
4. Najlepšia hodnota kvalitatívnej funkcie reprezentuje najlepšie zjednodušenie viacvrstvovej siete

Pri reálnych riešeniach je kvalitatívna funkcia často kompromis medzi zjednodušením a stratou informácie. Čím viac vrstiev by sme spojili, tým viac by sa nám sieť zjednodušila, ale zároveň by sa nám zväčšovala strata informácie, ktorá pri spájaní vzniká.

5.0.2 Lokálne zjednodušovanie

Druhým z prístupov zjednodušovania siete je lokálne zjednodušovanie. Na rozdiel od globálneho zjednodušovania je cieľom zníženie šumu generovaného priveľkým počtom vrstiev. Počiatočná idea prístupu vychádza z predpokladu, že čo je relevantné v sociálnej sieti má byť pochopiteľné v rámci relačnej - dyadickej - perspektívy a čo je relevantné je rozdielne v rozdielnych častiach jednej vrstvy.

Môžeme si to vysvetliť na AUCS sieti s tým, že vrstva bude pre nás lokálne relevantná vtedy, ak je to jediná vrstva umožňujúca prepojiť špecifického aktéra s veľkou častou jeho susedov. V takomto prípade v AUCS oddelení je vrstva obedov relevantná, pretože obedné prestávky sú

často jediným časom, kedy sa zamestnanci môžu stretnúť. Pre inú skupinu vedcov, môže byť takouto vrstvou facebook, kvôli faktu, že nepracujú v tej istej budove.

Z týchto údajov nám vyplýva, že relevantnosť jednotlivých vrstiev nevyplýva z ich globálnych informácií, ale z vyhodnotenia relevancie na lokálnej úrovni. Použitím takéhoto druhu zjednodušenia nám stále ostáva viacvrstvová sieť, ale s odstráneným šumom.

Formálny koncept relevancie bol zavedený Rossim a Magnanim [20] a je ho možné vyjadriť v rôznych metrikách, podľa toho, aké vzory nám majú z dát vzniknúť.

Lokálne zjednodušenie môžeme vyjadriť v nasledujúcich krokoch:

1. Zvolíme metriku $r(a, l)$ indikujúcu relevantnosť aktéra a na vrstve l a prah Θ.
2. Pre každú vrstvu l a pre každý pár aktérov a_1, a_2 ponecháme medzi nimi hranu na tejto vrstve, ak predtým hrana existovala a $r(a_1, l) \geq \Theta$ a zároveň $r(a_2, l) \geq \Theta$.

5.1 Projekcia

Metóda projekcie je tradičným prístupom ako zjednodušovať dvoj-módové siete, ktoré môžu byť modelované ako viacvrstvové siete, kde každý druh vrcholov je reprezentovaný vlastnou vrstvou.

Najjednoduchšou projekciou je odstránenie uzlov iných typov a presun ich hrán medzi typ vrcholov, ktorý ponechávame. Tento prístup má viacero nevýhod, ktorými sú odstránenie informácie o vrcholoch, ktoré sú prepojené k iným vrcholom rovnakého typu a druhou nevýhodou takého prístupu je generovanie veľkého počtu klík.

Spôsob, ako predísť týmto problémom je využitie váženej projekcie. Pri tomto druhu projekcie je hrana prepojená s váhou w definovanou ako $w(i, j) = \sum_p 1$, kde p označuje uzly ktoré sú iného typu a sú prepojené k i a j.

5.2 Splošťovanie

Ďalšou metódou zjednodušenia siete je splošťovanie (ang. flattening). Odlišuje sa od projekcie tým, že uzly na rozličných vrstvách reprezentujú rovnakých aktérov. Pri projekcii sa zjednodušuje sieť, ktorá obsahuje viaceré druhy uzlov.

Základný druh splošťovania sa skladá z vytvorenia jednej vrstvy, ktorá obsahuje všetkých aktérov a hrany medzi nimi z ostatných vrstiev siete.

Definícia 3 *Základné (nevážené) splošťovanie viacvrstvovej siete* $G = (A, L, V, E)$ *je graf* (V_f, E_f), *kde* $V_f = \{a|(a, l) \in V\}$ *a* $E_f = \{(a_i, a_j)|\{(a_i, l_q), (a_j, l_r)\} \in E\}$.

Jednoduchou variáciou základného splošťovania nám vzniká váženie splošťovanie, kde sa ku každej hrane pridá váha, ktorá zodpovedá počtu hrán medzi rovnakými aktérmi vo viacvrstvovej sieti.

Ďalším obecnejším prístupom je priradenie váhy $\Theta_{q,r}$ každému páru vrstiev (l_q, l_r), čo nám umožňuje vyjadriť jednovrstvovú sieť ako lineárnu kombináciu pôvodnej viacvrstvovej siete.

Definícia 4 *Máme viacvrstvovú sieť* $G = (A, L, V, E)$. *Ak máme maticu* $\Theta = |L| \times |L|$, *kde* $\Theta_{q,r}$ *reprezentuje váhu priradenú hrane z vrstvy* l_q *k vrstve* l_r, *potom vážené sploštenie* G *definujeme ako vážený graf* (V_f, E_f, ω), *kde* (V_f, E_f) *je základné sploštenie* G *a* $\omega(a_i, a_j) = \sum_{\{(a_i,l_q),(a_j,l_r)\}\in E} \Theta_{q_r}$.

6 Detekcia komunít

Cieľom detekcie komunít je nájsť skupiny uzlov (v prípade viacvrstvových sietí aktérov), ktoré sú navzájom hustejšie prepojené oproti zvyšku siete. Takéto skupiny majú často spoločné vlastnosti, ktoré ich oddeľujú od zvyšku siete. V prípade reálnych sietí môžu byť týmito skupinami rodiny, kolegovia z kancelárie, spolužiaci v triede, v prípade internetu sú to napríklad webové stránky zaoberajú sa podobnou tematikou.

Wasserman a Faust [21] identifikovali tri základné idey, ktoré sa uplatňujú pri identifikácií skupín, sú nimi: *dosažitelnosť*, *susedstvo* a *vrcholový stupeň*. Z týchto ideí vyplývajú štyri základné obecné vlastnosti, ktoré ovplyvnili väčšinu formalizácií konceptov detekcie komunít, sú to:

- Vzájomnosť prepojení
- Blízkosť alebo dosažitelnosť členov skupín
- Frekvencia prepojení medzi členmi
- Relatívna frekvencia prepojení medzi členmi skupín v porovnaní s ostatnými

Detekcia komunít má aj svoje praktické použitie. Využitím detekcie komunít na sieti webových stránok a geografických lokácií ich užívateľov môžeme získať informácie o skupinách klientov, ktorí pristupujú k rovnakým stránkam z podobnej geografickej lokácie. S využitím zrkadlených serverov, ktoré sú umiestnené bližšie k užívateľom, sme schopní vylepšiť výkon stránok [22]. Identifikácia skupín zákazníkov v sieti relácie medzi zákazníkmi a nakúpeným tovarom u online predajcov, nám umožňuje vytvorenie efektívneho systému odporúčaní [23].

Aplikácie detekcie komunít sú dôležité aj z iného pohľadu. Pri analýze vnútornej štruktúry komunity a aktérov/vrcholov môžeme odhaliť vrcholy, ktoré sú centrálnejšie v rámci komunity tým, že majú väčší počet spojov s ostatnými členmi komunity. Takéto vrcholy sú dôležité z pohľadu stability skupiny a často sú aj prepojením na ďalšie komunity.

6.1 Komunita

Pred tým ako si ukážeme niektoré prístupy k detekcii komunít, si zadefinujeme, čo pre nás pojem komunita znamená. Neexistuje žiaľ žiadna univerzálne akceptovateľná definícia, tá často podlieha konkrétnemu riešenému problému. Našťastie väčšina týchto definícií ma spoločný prvok, a to fakt, že komunity by mali mať viacej prepojení v rámci svojej skupiny ako do svojho okolia. Definícia, ktorú nižšie priblížim pochádza od Santo Fortunato [4] a volím ju z pohľadu jej obecnosti a použitia vyššie spomenutého faktu o počte prepojení.

Začíname s podgrafom C grafu G, kde počty $|C| = n_c$ a $|G| = n$ reprezentujú počet vrcholov. Zadefinujeme si vnútorný stupeň vrcholu $v \in C$, k_v^{int}, ktorý určuje počet hrán spájajúcich vrchol v s ostatnými vrcholmi v C. Podobne si zadefinujeme vonkajší stupeň vrcholu $v \in C$, k_v^{ext}, ktorý určuje počet hrán spájajúcich vrchol v s vrcholmi vo zvyšku grafu. Ak sa $k_v^{ext} = 0$, má vrchol

v hrany len v rámci komunity C a môžeme ho považovať za dobrý vrchol z pohľadu komunity. V prípade, že $k_v^{int} = 0$ vrchol nemá ani jednu hranu, ktorá by ho spájala s komunitou C, je vhodné ho umiestniť do inej komunity.

Vnútorný stupeň k_{int}^C komunity C je suma vnútorných stupňov vrcholov v C. Vonkajší stupeň k_{ext}^C komunity C je suma vonkajších stupňov vrcholov v C. Celkový stupeň k^C je sumou stupňov všetkých vrcholov v C.

Zadefinujme si vnútro-zhlukovú hustotu $\delta(C)$ podgrafu C ako pomer medzi počtom vnútorných hrán C a počtom všetkých možných vnútorných prepojení.

Definícia 5 $\delta_{int}(C) = \frac{\textit{počet vnútorných hrán } C}{n_c(n_c-1)/2}$

Podobne si zadefinujeme aj vonkajšiu-zhlukovú hustotu, $\delta_{ext}(C)$ je pomer medzi počtom hrán medzi vrcholmi z C do zvyšku grafu a maximálnym počtom medzi-zhlukových hrán.

Definícia 6 $\delta_{ext}(C) = \frac{\textit{počet medzi-komunitných hrán } C}{n_c(n-n_c)}$

Preto, aby C bola komunitou predpokladáme, že $\delta_{int}(C)$ je vnímateľne väčšia ako priemerná hustota prepojení $\delta(G)$, ktorá je definovaná ako pomer medzi počtom hrán v G a maximálnym počtom hrán $n(n-1)/2$. Na druhú stranu pri $\delta_{ext}(C)$ predpokladáme, že bude výrazne nižšie ako $\delta(G)$.

Jednou z potrebných vlastností komunity je prepojenosť. Predpokladáme, že existuje cesta medzi každým párom vrcholov z C a to len medzi vrcholmi z C. Táto vlastnosť nám umožňuje zjednodušiť detekciu komunít pre nesúvislé grafy tým, že môžeme detekciu komunít aplikovať na každú súvislú komunitu zvlášť.

Definíciu komunity môžeme rozdeliť v troch rozdielnych kontextoch, na lokálnu, globálnu a na definíciu založenú na podobnosti vrcholov.

6.1.1 Lokálna definícia

Lokálne definície sa zameriavajú na podgrafy, ktoré študujú, prípadne ich susedstva. Jednou z takýchto definícii je klika.

Definícia 7 *Klikou v neorientovanom grafe $G = (V, E)$ chápeme ako podmnožinu vektorov $C \subseteq V$, kde medzi každými dvoma vrcholmi existuje hrana.*

Táto definícia je často nevyhovujúca z dôvodu, že je veľmi reštriktívna. Menej reštriktívnou variáciou je n-klika.

Definícia 8 *N-klikou v neorientovanom grafe $G = (V, E)$ chápeme ako podmnožinu vektorov $C \subseteq V$, kde vzdialenosť medzi každými dvoma vrcholmi je nanajvýš n.*

Existujú ešte varianty ako n-klan a n-klub. N-klan je n-klika, ktorej priemer nie je väčší ako n. N-klub je je maximálny podgraf o priemere n.

Definícia 9 *Silná komunita je podgraf, ktorého vnútorný stupeň každého vrcholu je väčší ako vonkajší stupeň vrcholu.*

Definícia 10 *Slabá komunita je podgraf, ktorého vnútorný stupeň podgrafu je väčší ako vonkajší stupeň.*

6.1.2 Globálne definície

Definície vychádzajú z pohľadu na celkovú štruktúru grafu. Táto definícia je vhodná v prípadoch, kde zhluky sú esenciálne časti grafu, ktoré nemôžu byť oddelené bez toho, aby výrazne ovplyvnili funkčnosť systému. Tieto definície sú často nepriame a vychádzajú z globálnych vlastností grafu použitých pri algoritmoch detekcie komunít. Jednou z priamych globálnych definícií je *null model*, teda graf, ktorý je štrukturálne identický s pôvodným grafom, ale inak je náhodný. *Null model* je použitý na porovnanie a verifikáciu toho, či originálny graf obsahuje komunitnú štruktúru alebo nie.

Najpopulárnejším null modelom je model od Newman-a a Girvan-a [24], ktorý sa skladá z náhodnej verzie pôvodného grafu, v ktorom sú hrany poprepájané náhodne pod podmienkou, že sa pôvodný stupeň vrcholu nezmení v náhodnom grafe. Z tohto null modelu vychádza definícia modularity, ktorej sa budem venovať v kapitole 7.

6.1.3 Definície založené na podobnosti vrcholov

Je prirodzené predpokladať, že komunity budú obsahovať vrcholy, ktoré sú si podobné. Komunity sú v takýchto prípadoch vytvorené nasledovne:

- Pre každý pár vrcholov je vypočítaná ich podobnosť
- Komunity sú vytvorené z vrcholov, ktoré sú si najpodobnejšie

Ako miery podobnosti sú často používané vzdialenosti medzi vrcholmi, či už je to euklidovská vzdialenosť, manhattan, kosinová podobnosť a iné. Ďalší metódou je meranie prekrytia susedov jednotlivých vrcholov. Štrukturálnou mierou, ktorá môže byť použitá, je Pearsonova korelácia medzi stĺpcami a riadkami matice susednosti.

6.2 Detekcia na jednovrstvových sieťach

V nasledujúcich sekciách priblížim metódy detekcie komunít, ktoré sú určené na jednovrstvové siete. Sú dôležité aj z pohľadu viacvrstvových sietí, lebo nám umožňujú vykonať detekciu komunít na určitej zvolenej vrstve, ktorú sme si mohli zvoliť, napríklad na základe najväčšej podobnosti s ostatnými, alebo pri použití splošťovacích prístupov na viacvrstvovej sieti.

6.2.1 Tradičné metódy

Medzi tradičné metódy zahŕňame metódy a algoritmy, ktoré nie sú priamo určené na detekciu komunít. Často sú to algoritmy určené na zhlukovanie, poprípade delenie grafu.

Delenie grafu Základnou ideou prístupov delenia grafu je rozdelenie grafu na g skupín s predefinovanou veľkosťou, tak aby bol počet hrán medzi skupinami minimálny. Počet hrán, ktoré sa nachádzajú medzi skupinami sa nazýva *veľkosť rezu*. Veľkosť skupiny musí byť špecifikovaná. Pokiaľ by tomu tak nebolo, tak riešenie rezu by bolo triviálne (všetky vrcholy by skončili v jednej skupine).

Delenie grafu je základným problémom pri paralelnom programovaní, delení obvodov a v návrhu sériových algoritmov ako aj pri technikách riešiacich parciálne diferenciálne rovnice. Väčšina variant problému delenia grafu je však NP-ťažký problém.

Existuje viacero algoritmov, ktoré problém riešia, no riešenie nemusí byť optimálne. Príkladom algoritmov z tejto kategórie sú Kerningham-Lin algoritmus [25] a metóda spektrálnej bisekcie.

Hierarchické zhlukovanie Pri detekcii komunít často nepoznáme komunitnú štruktúru grafu, počet komunít nám je často neznámy. Metódy, ktoré musia mať dopredu daný počet komunít nám, nemusia vrátiť vždy najlepší výsledok. V takýchto prípadoch, ako aj v prípadoch, kde má graf hierarchickú štruktúru, je vhodné použiť algoritmy hierarchického zhlukovania. Pri hierarchickom zhlukovaní je nutné definovať mieru podobnosti medzi vrcholmi. Metódy založené na hierarchickom zhlukovaní delíme do dvoch skupín:

1. **Aglomeratívne algoritmy**, kde sa zhluky postupne spájajú, pokiaľ je ich podobnosť vysoká, prístup zdola nahor
2. **Divizívne algoritmy**, kde sa zhluky iteratívne delia a odstraňujú sa hrany spájajúce vrcholy s nízkou podobnosťou, prístup zhora nadol

Najväčšou výhodou takýchto prístupov je fakt, že nepotrebuje žiadnu informáciu o štruktúre grafu. Nevýhodami sú problémy s klasifikáciou, kde vrcholy s jedným susedom sa stávajú zhlukmi, čo nemusí byť v našom vnímaní správne. Taktiež nám vždy vznikne hierarchická štruktúra, aj pokiaľ dáta žiadnu takúto štruktúru neobsahujú a azda najväčším problémom je škálovateľnosť. Algoritmy v najlepších prípadoch dosahujú výpočtovej náročnosti $O(n^2)$.

Spektrálne zhlukovanie Metódy zahŕňajú všetky prístupy, kde sa k deleniu množiny používajú vlastné vektory matíc. Spektrálne zhlukovanie pozostáva z transformácie pôvodnej množiny do priestoru, kde koordináty prvkov sú ich vlastné vektory. Tieto body v priestore sú následné zhlukované iný metódami. Medzi prvé prístupy spektrálneho zhlukovania bolo využitie vlastných vektorov matice susednosti Donath a Hoffmanom [26] na delenie grafu. Najčastejšie používanou

maticou je Laplasiánska matica a na základe toho, či je normalizovaná delíme metódy na *nenormalizované spektrálne zhlukovanie* a *normalizované spektrálne zhlukovanie*.

6.2.2 Divizivné algoritmy

Ďalšou skupinou algoritmov sú divizívne algoritmy. Základnou myšlienkou takýchto algoritmov je nájdenie a následné odstránenie hrán, ktoré spájajú rôzne komunity. Divizívne metódy nám neponúkajú výraznejšie konceptuálne zlepšenie oproti tradičným metódam, dané tým, že vykonávajú vo svojej podstate hierarchické zhlukovanie. Najzákladanejším rozdielom je odstraňovanie hrán medzi zhlukmi oproti odstraňovaniu hrán medzi vrcholmi s najmenšou podobnosťou.

Girvan a Newman metóda Najpopulárnejším algoritmom v oblasti divizívnych algoritmoch je algoritmus Girvan-a a Newman-a [27]. Metóda je dôležitá z historického hľadiska, bola priekopnícka v oblasti detekcie komunít. Hrany sú vyberané na základe mier centrality hrán, ktoré určujú dôležitosť hrany na základe nejakej vlastnosti. Algoritmus sa skladá z nasledovných krokov:

1. Vypočítanie centrality pre všetky hrany.
2. Odstránenie hrany s najväčšou centralitou. V prípade, ak máme na výber z viacerých hrán, jednu z nich vyberieme náhodne.
3. Prepočítanie centralít na novom grafe.
4. Opakujeme v cykle kroky 2 a 3.

Girvan a Newman ako mieru použili betweenness reprezentujúcu frekvenciu účasti hrany na procese. Používali tri alternatívne definície:

Betweenness hrán je počet najkratších hrán medzi všetkými pármi vrcholov, ktoré prechádzajú danou hranou.

Betweenness náhodnej prechádzky je určená počtom prechodov náhodného chodca cez hranu počas náhodnej prechádzky.

Betweenness aktuálneho prúdu je založená na myšlienke, kde graf predstavuje sieť rezistorov a hrany majú jednotky rezistencie. Pre každú hranu je vypočítaná hodnota prúdu na základe Kirchoffových rovníc. Betweenness je určená priemerným prúdom, ktorý hrana nesie.

Algoritmus vygeneruje dendrogram, pre určenie najlepšieho delenia sa využíva modularita, ktorú si zadefinujeme nižšie.

Existujú modifikácie tejto metódy, ktoré sú rýchlejšie. Modifikácia navrhnutá Tyler-om a ostatnými [28] zlepšuje výkon tým, že miesto výpočtu betweenness hrán pomocou každého vrcholu,

sa vezme len časť náhodne vybraných vrcholov, zvaných centrá. Miesto modularity sa používa overenie špecifikovanej definície komunity, ktorá je v tomto prípade definovaná ako súvislý podgraf s n_0 vrcholmi, kde betweenness každej hrany v komunite nepresahuje hodnotu $n_0 - 1$. Ďalšia metóda od Rattigan-a a ostatných [29] využíva rýchlu aproximáciu betweenness hrán pomocou sieťového štrukturálneho indexu, ktorý sa skladá z anotácií vrcholov v kombinácií so vzdialenostnou metrikou. Výpočtová náročnosť sa týmito modifikáciami znížila na $O(m)$.

6.2.3 Metódy založené na modularite

Modularita, pôvodne navrhnutá ako zastavujúce kritérium pre Newman-Girvan algoritmus, sa stala hlavnou časťou mnohých algoritmov. Je najpoužívanejšou a najznámejšou kvalitatívnou funkciou. Budeme vychádzať z modularity definovanej v sekcii 7.1. Najlepšie rozdelenie grafu nastáva pri vysokých hodnotách modularity.

Optimalizácia modularity Pri metódach, ktoré optimalizujú modularitu sa snažíme dosiahnuť čo najvyššiu hodnotu, maximalizujeme modularitu. Metódy patriace do tejto skupiny sú jedni z najviac používaných. Hoci dosiahnutie maxima modularity je NP-ťažký problém, existujú riešenia, ktoré prinášajú dostatočne dobré aproximácie v rozumnom čase.

Jedným z prvých algoritmov je greedy verzia Girvan-Newman algoritmu. Začíname len s vrcholmi pôvodného grafu a postupne pridávame hrany na základe najlepšej modularity.

Ďalšou metódou je využitie simulovaného žíhania, pravdepodobnostného algoritmu, navrhnutú Guimerà-om [30], kde sa optimalizuje modularita na základe lokálnych (jeden vrchol zmení svoju komunitu) a globálnych (spájajú a delia sa komunity) zmien.

Louvain Heuristický algoritmus [8] založený na optimalizácii modularity. Najväčšou výhodou tohto algoritmu je výpočtový výkon. Algoritmus je rozdelený na dve fáze:

Fáza 1 Na začiatok priradíme každému vrcholu vlastnú komunitu, následne pre každý vrchol i ohodnotíme nárast modularity u susedov j, tým, že by sme presunuli vrchol i do komunity vrcholu j. Následne je vrchol i presunutý do komunity vrcholu j s najväčším ziskom modularity. Tento proces je opakovaný dokiaľ už nie je možné zlepšenie. Koncom prvej fáze sa modularita dostáva do lokálneho maxima.

Fáza 2 Skladá sa z vytvorenia novej siete, kde komunity sú nové vrcholy grafu, hrany medzi nimi sú vytvorené na základe hrán medzi vrcholmi v komunitách, pričom sa váhy týchto hrán sčítajú. Po dokončení druhej fázy sa presúvame späť do prvej fázy s novým grafom až pokiaľ nám nevznikne graf s jedným vrcholom.

6.2.4 Spektrálne algoritmy

Do tejto skupiny algoritmov patria prístupy založené na spektrálnych vlastnostiach matíc grafov. Medzi takéto prístupy patrí spektrálne zhlukovanie grafov s využitím vlastných vektorov Lapla-

siánskej matice ako aj optimalizácia modularity Girvan-Newman s vyžitím vlastných vektorov matice modularity.

6.3 Detekcia na viacerých vrstvách

Existuje niekoľko prístupov, akým sa dá aplikovať detekcia komunít na viacvrstvových sieťach s plných využitím jej štruktúry.

6.4 Consensus Clustering

Prvý z prístupov detekcie komunít s využitím všetkých dimenzií siete je Consensus Clustering. Základným princípom tejto metódy je aplikácia tradičných algoritmov na detekciu komunít na jednotlivé vrstvy a následná agregácia ich výsledkov. Vychádzame z predpokladu, že vrcholy, ktoré sú často v spoločnej komunite na jednotlivých vrstvách, majú väčšiu pravdepodobnosť byť v spoločnej komunite aj z pohľadu viacvrstvovej siete.

Dolovanie frekventovaných položkových množín Na získanie komunít sa v tejto skupine metód využíva princíp dolovania frekventovaných položkových množín. Vrcholy vo viacvrstvovej sieti definujú transakcie a položky sú dvojice (c, d), kde c reprezentuje komunitu a d dimenziu do ktorej vrchol patrí. Napríklad, máme vrchol v_i a komunity c_1, c_5 a c_7 v dimenziách d_1,d_2, d_3 do ktorých vrchol patrí. I-ta transakcia je množina položiek $\{(c_1, d_1), (c_5, d_2), (c_2, d_3)\}$. Následne na tieto transakcie použijeme niektorý z algoritmov na dolovanie frekventovaných položiek. Výsledné frekventované položky reprezentujú komunity daných vrcholov. Príkladom takého algoritmu je ABACUS [6].

Algoritmus založený na podobnosti zhlukov Princíp toho algoritmu je nasledovný. Pre každý pár vrcholov sa vypočíta podobnosť, ktorá je založená na počte komunít, napríklad, ak máme viacvrstvovú sieť s piatimi vrstvami a vrcholy v_i a v_j sú v spoločnej komunite na troch vrstvách, tak výsledná podobnosť týchto vrcholov (v_i, v_j) je 3/5.

Po tom, čo je vypočítaná podobnosť pre všetky páry vrcholov, je použitý nejaký zvolený algoritmus zhlukovania (napr. K-Means), ktorý na základe vypočítanej podobnosti vrcholov vytvorí nové komunity. Tento proces je známy aj ako integrácia partícií.

Generalizované kanonické korelácie Všetkým vrcholom v jednotlivých komunitách je priradený pridelený bod v l-dimenzionálnom euklidovskom priestore. Čím je cesta medzi vrcholmi kratšia, tým sú so body bližšie v priestore. Jedno z takýchto mapovaní je spojenie vrchných vlastných vektorov matice susednosti, v prípade d dimenzionálnej siete nám vznikajú matice vlastností S_i veľkosti $l \times n$ kde stĺpec matice je pozícia vrcholu v l-dimenzionálnom priestore. Následne sú tieto matice agregované pomocou lineárnych transformácií tak, aby sa maximalizovali párové korelácie jednotlivých S_i matíc. Využíva sa práve metóda generalizovaných kanonických korelácií [31]. Následne je na agregovanej matici použitý algoritmus na zhlukovanie.

6.5 Detekcia mostov

Mostom v oblasti teórie grafov je hrana s vysokým informačným tokom. Z pohľadu detekcie komunít sú tieto hrany dôležité, lebo ich odstránením nám často vznikajú oddelené komunity.

CLECC Community Detection V sociálnych sieťach je vhodné mať silné spoje medzi komunitami. Na identifikáciu slabých spojov medzi pármi vrcholov navrhol Brodka mieru *CLECC* [7].

Definícia 11 (Viacvrstvové susedstvo) *$MN(x, \alpha)$, určitého vrcholu $x \in V$ je množina vrcholov, ktoré sú susedmi vrcholu x aspoň na α vrstvách.*

$$MN(x, \alpha) = \{y : |\{l :< x, y, l >\in E \vee < y, x, l >\in E\}| \geq \alpha\} \tag{1}$$

Definícia 12 (Medzivrstvový zhlukovací koeficient hrany) *pre hranu $< x, y >$ vyjadruje prepojenosť susedov vo viacvrstvových sieťach.*

$$CLECC(x, y, \alpha) = \frac{|MN(x, \alpha) \cap MN(y, \alpha)|}{|(MN(x, \alpha) \cup MN(y, \alpha))/\{x, y\}|} \tag{2}$$

Inak povedané reprezentuje pomer medzi počtom spoločnými viacvrstvovými susedami a všetkými viacvrstvovými susedami.

Následne algoritmus funguje na podobnom princípe ako algoritmus Girvan-Newman, jeho kroky sú nasledovné:

1. Vypočítame $CLECC(x, y, \alpha)$ pre každý pár (x, y), kde $x \in MN(y)$ a zvolené α.
2. Odstránime všetky hrany medzi párom (x, y), kde je CLECC hodnota najnižšia. Ak je viacej párov s najnižšou hodnotou CLECC, je vybraný jeden pár z nich náhodne.
3. Prepočítame $CLECC(x, y, \alpha)$ pre všetky $z : z \in MN(x) \cup MN(y)$ a zvolené α.
4. Ak odstránenie hrán viedlo k rozdelení siete do podgrafov, overíme podgrafy oproti dopredu definovanej podmienke pre existenciu komunity. Ak dosiahneme požadovaný počet komunít už ďalej hrany neodstraňujeme.
5. Opakujeme krok 2 dokiaľ nám neostanú len komunity alebo osamotené vrcholy.

7 Ohodnotenie výsledkov detekcie komunít

V nasledujúcej kapitole si priblížime ako vyhodnocovať získané komunity z procesu detekcie komunít. Tieto ohodnotenia nám pomôžu určiť či komunity, ktoré sme získali, sú dobre oddelené.

7.1 Ohodnotenie na jednej vrstve

Modularita

Predpokladajme, že máme sieť s N vrcholmi a L hranami rozdelenú do n_c komunít, každá komunita má N_c vrcholov spojených k ostatným v komunite L_c hranami. *Modularita* nám meria rozdiel oproti skutočnému diagramu prepojenia (A_{ij}) a predpokladanému počtu hrán medzi i a j v prípade, že by sieť bola náhodná.

$$M = \sum_{c=1}^{n_c} \left[\frac{L_c}{L} - \left(\frac{k_c}{2L} \right)^2 \right] \tag{3}$$

Pokrytie

Pod pokrytím rozumieme pomer počtu vnútro-komunitných hrán k celkovému počtu hrán. V ideálnom prípade, kde by komunity neboli navzájom prepojené hranami, by pokrytie nadobúdalo hodnotu jedna.

$$\mathcal{C}(P) = \frac{|e : (i,j) \in E \wedge i \in C_i \wedge j \in C_i\|}{|E|} \tag{4}$$

Výkon

Kvalitatívna funkcia výkon $\mathcal{P}$ určuje počet dobre ïnterpretovaných"párov vrcholov, t.j. dva vrcholy patriace do komunity a prepojené hranou alebo dva vrcholy z rôznych komunít a neprepojené hranou.

$$\mathcal{P}(P) = \frac{|\{(i,j) \in E, C_i = C_j\}| + |\{(i,j) \notin E, C_i \neq C_j\}|}{n(n-1)/2} \tag{5}$$

Výkon nadobúda hodnôt v rozmedzí [0, 1].

7.2 Ohodnotenie na viacerých vrstvách

Ohodnotenia na viacerých vrstvách berú do úvah aj viacvrstvovú štruktúru sietí.

Rozmanitosť

Určuje v koľkých dimenziách je komunita vyjadrená. Je určená pomerom počtu dimenzií vyjadrených v komunite a celkového počtu komunít.

$$\mathcal{V}_c = \frac{|D_c| - 1}{|D| - 1} \tag{6}$$

Výsledná hodnota je v rozsahu [0,1], kde hodnota jeden reprezentuje fakt, že komunita je vyjadrená vo všetkých dimenziách, hodnota nula vyjadruje fakt, že komunita je vyjadrená len v jednej dimenzii. Rozmanitosť nie je definovaná pre $|D| = 1$, čo môže nastať len pri jednodimenzionálnych sieťach.

Exkluzivita

Určuje nám koľko párov aktérov v komunite je prepojených len v jednej dimenzii. Môže byť vypočítaná ako pomer medzi exkluzívnymi spojeniami v komunite a celkovým počtom prepojených párov v komunite. Exkluzívnym spojením chápeme spojenie medzi dvoma aktérmi, kde existuje len jedna dimenzia, na ktorej by bola hrana, ktorá by aktérov spájala.

$$\mathcal{E}_c = \frac{\sum_{d \in D} |\overline{P_{c,d}}|}{|P_c|} \tag{7}$$

Exkluzivita je rovná nule, ak neexistujú v komunite žiadne exkluzívne spojenia, to znamená, že každý pár aktérov je spojený buď na žiadnej alebo na viac ako jednej vrstve. Je rovná jednej ak všetky spojenia sú exkluzívne, t.j. všetky páry aktérov sú spojené nanajvýš na jednej vrstve.

Homogenita

Určuje ako uniformné je rozloženie hrán naprieč dimenziami v komunite. K výpočtu potrebujeme vedieť štandardnú odchýlku distribúcie hrán v komunite na dimenziách σ_c

$$\sigma_c = \sqrt{\frac{\sum_{d \in D}(|P_{c,d}| - avg_c)^2}{|D|}} \tag{8}$$

kde avg_c je priemer distribúcie, ako aj maximálnu teoretickú štandardnú odchýlku σ_c^{max}.

$$\sigma_c^{max} = \sqrt{\frac{(max(|P_{c,d}|) - 1)^2}{2}} \tag{9}$$

kde $max(|P_{c,d}|)$ je počet hrán patriacich k najviac reprezentovanej dimenzii v komunite. Homogenita je vyjadrená nasledovne:

$$\mathcal{H}_c = 1 - \frac{\sigma_c}{\sigma_c^{max}} \tag{10}$$

Hodnota je rovná jednej pokiaľ sú hrany distribuované rovnomerne.

Komplementarita

Je určená súčtom rozmanitosti, exkluzivity a homogenity.

$$\gamma_c = \mathcal{V}_c \times \mathcal{E}_c \times \mathcal{H}_c \tag{11}$$

Redundancia

Zachycuje fenomén, kde množina aktérov z ktorých sa skladá komunita na určitej dimenzii, tvoria komunitu aj na iných dimenziách. Indikuje nám redundanciu spojení. Čím väčší počet dimenzií nám spája páry aktérov, tým vyššia bude aj redundancia.

$$\rho_c = \sum_{(u,v \in \overline{P_c})} \frac{|\{d : \exists (u,v,d) \in E\}|}{|D| \times |P_c|} \tag{12}$$

8 Úvod do praktickej časti

Praktická časť mojej diplomovej práce sa skladá z implementácie vybraných metód na detekciu komunít na viacvrstvových sieťach. Na analýzu dát som vytvoril knižnicu s implementovanými algoritmami a následne webovú aplikáciu, ktorá túto knižnicu využíva a umožňuje užívateľom analyzovať viacvrstvové siete z pohľadu komunít a porovnávať výsledky jednotlivých algoritmov. Na vizualizáciu dát a výsledkov som naprogramoval samostatnú webovú službu, ktorú následne hlavná webová aplikácii prevoláva.

Pri implementácií algoritmov som sa riadil teoretickými základmi popísanými v predošlých kapitolách, implementáciami z odlišných knižníc ako aj rôznymi článkami zaoberajúcimi sa danou tematikou.

Zdrojový kód je voľne dostupný na GitHub-e[1], implementovaná knižnica je dostupná na `https://github.com/matejkubinec/mncd`, webová aplikácia na analýzu na `https://github.com/matejkubinec/mncd-app` a webová služba je dostupná na `https://github.com/matejkubinec/mncd-viz`.

Okrem repozitárov je webová aplikácia nasadená aj na `https://mncd.azurewebsites.net`.

8.1 Vízia

Víziou projektu je umožnenie užívateľovi analyzovať viacvrstvové siete z pohľadu detekcie komunít. Užívateľ by mal mať možnosť zvoliť prístup detekcie komunít, či už sa jedná výber jednej vrstvy, sploštenie, poprípade analýza z pohľadu viacvrstvových sietí. Mal by mať na výber z viacerých algoritmov implementujúcich tieto prístupy a mal by mať možnosť zvoliť ich parametre. Dáta, ktoré sa analyzujú by mali byť dostupné, ale užívateľovi by malo byť umožnené analyzovať aj vlastné dáta. Výsledky týchto analýz by mali byť prehľadne dostupné užívateľovi a mal by mať možnosť výsledky jednotlivých analýz medzi sebou porovnávať.

Samotná implementácia algoritmov by nemala byť závislá na aplikácií, malo by byť umožnené využiť implementácie aj v iných projektoch.

8.2 Existujúce riešenia

V tejto podkapitole priblížim už existujúce riešenia, ktoré sa zaoberajú problematikou viacvrstvových sietí a detekcie komunít.

MuxViz [32] je framework pre analýzu a vizualizáciu viacvrstvových sietí, umožňuje interaktívnu vizualizáciu a prieskum viacvrstvových sietí. Okrem detekcie komunít umožňuje aj prácu s korelačnými metrikami, vizualizáciu viacvrstvových sietí, geografických viacvrstvových sietí ako aj vizualizáciu dynamických aspektov viacvrstvových sietí. Implementovaný je De Domenicom v R, a v GNU Octave.

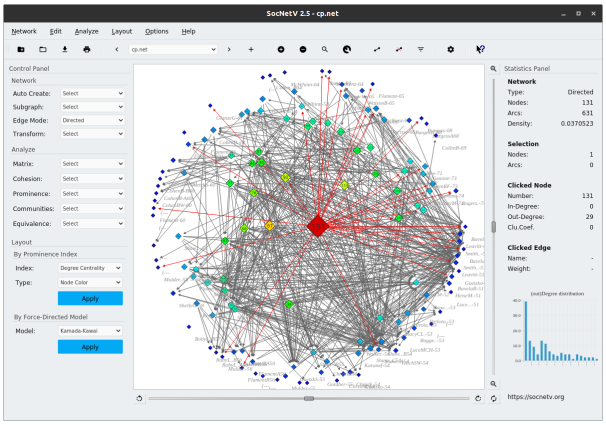
[1] GitHub je webová služba slúžiaca na hosting zdrojového kódu, dostupná na `https://github.com`.

multiNetX [33] je python balíček na manipuláciu a vizualizáciu viacvrstvových sietí. Balíček je postavený nad známym balíkom na prácu s grafmi NetworkX. Knižnica umožňuje vytváranie vážených a neválených neorientovaných viacvrstvových sietí, analýzu spektrálnych vlastností matice susednosti a Laplasiánskej matice a vizualizáciu dynamických procesov cez vyfarbenie hrán a vrcholov.

Py3Plex [34] je python balíček na prácu s viacvrstvovými sieťami. Ponúka základnú množinu algoritmov pre štatistickú analýzu takýchto sietí, kolekciu algoritmov na dekompozíciu a obaľuje aj vysoko efektívne implementácie algoritmov v pythonu. Okrem týchto funkcií ponúka aj možnosť vizualizácie.

NetworkX [35] je python balíček určený na manipuláciu a štúdium štruktúry, dynamických vlastností a funkcií komplexných sietí. Ponúka širokú škálu algoritmov ako aj vizualizácií. Aplikácia je však určená len pre jednovrstvové siete.

igraph [36] je kolekcia nástrojov na analýzu sietí s dôrazom na efektívnosť, prenosnosť a použiteľnosť. Môže byť využitá v jazykoch R, python, Mathematica a C/C++. Taktiež je to knižnica určená len na jednovrstvové siete.



Obr. 3: SocNetV (Zdroj: `socnetv.org`)

SocNetV (Social Network Visualizer) je medzi-platformná aplikácia určená na analýzu a vizualizáciu sociálnych sietí. Aplikácia obsahuje množstvo metrík a vizualizácií. Aplikácia je však určená len na tradičné siete.

Z prehľadu existujúcich riešení je jasné, že knižníc a aplikácií, ktoré sa venujú priamo problematike viacvrstvových sietí je málo a väčšina týchto knižníc obsahuje len malé množstvo implementovaných algoritmov a postupov. V čom tieto knižnice vynikajú sú vizualizačné prostriedky a preto som niektoré z nich využil aj vo svojej práci.

9 Knižnica MNCD

Hlavnou časťou praktickej časti je knižnica MNCD. Táto knižnica je napísaná v jazyku C# a spĺňa .NET štandard 2.1, čím je ju možné použiť v .NET Core 3.0 a vyššie. Slúži primárne na detekciu komunít vo viacvrstvových sieťach, ale je použiteľná aj pri detekcii komunít na jednovrstvových sieťach. Okrem algoritmov knižnica obsahuje aj miery ohodnotenia výsledných komunít, ktoré boli spomenuté v teoretickej časti, konkrétne v kapitole 7. V nasledujúcich podkapitolách sa budem venovať štruktúre a návrhu knižnice ako aj algoritmom, ktoré implementuje.

9.1 Prehľad projektu

Knižnica sa skladá z dvoch projektov **MNCD** a **MNCD.Tests**. **MNCD.Tests** obsahuje testy k algoritmom z projektu MNCD. MNCD sa skladá z nasledujúcich častí:

MNCD.Clique - Obsahuje algoritmy na detekciu klík v sieti, aktuálne obsahuje algoritmus *BronKerbosh* na výpočet maximálnych klík.

MNCD.CommunityDetection - obsahuje algoritmy detekcie komunít.

MNCD.Components - obsahuje algoritmus na výpočet súvislých komponent siete.

MNCD.Core - obsahuje doménové triedy.

MNCD.Evaluation - obsahuje metódy na ohodnotenie detekcie komunít.

MNCD.Extensions - obsahuje pomocné extension metódy.

MNCD.Flattening - obsahuje metódy na sploštenie viacvrstvových sietí.

MNCD.Generators - obsahuje algoritmy na generovanie sietí.

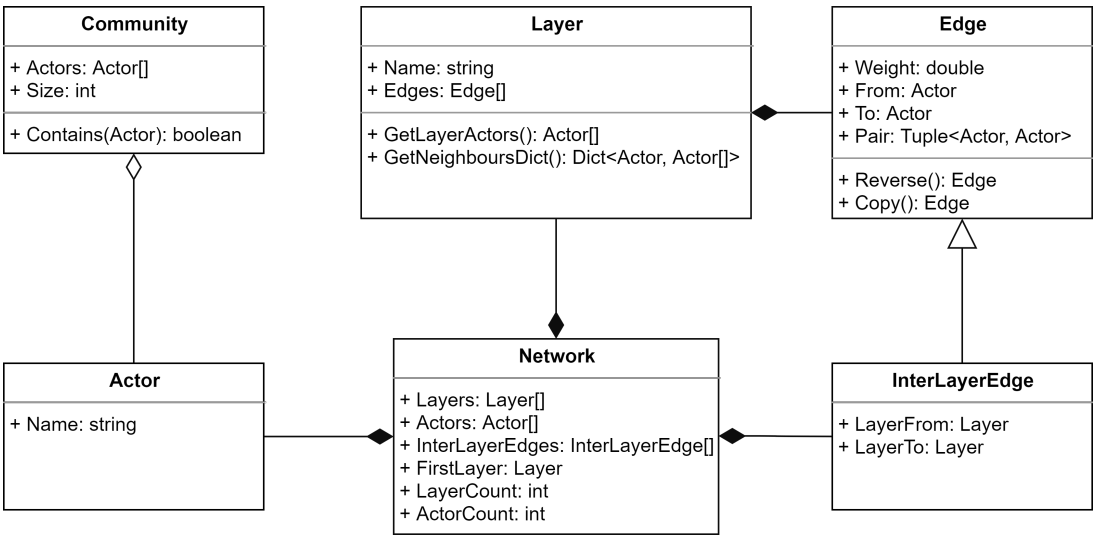
MNCD.Measures - obsahuje miery, či už pre hrany alebo vrcholy, napr. CLECC.

MNCD.Neighbourhood - obsahuje algoritmy pre výpočet susedstva.

MNCD.Readers - obsahuje triedy na čítanie siete zo súboru vo formáte MPX alebo zoznamu hrán.

MNCD.Writers - obsahuje triedy na zápis siete do súboru vo formáte zoznamu hrán.

9.2 Doména



Obr. 4: Triedny diagram domény knižnice MNCD

Na obrázku 4 môžeme vidieť triedny diagram doménových tried projektu.

Actor - trieda, ktorá reprezentuje aktéra vo viacvrstvovej sieti.

Edge - trieda reprezentujúca hranu, obsahuje aktérov, ktorých hrana spája a váhu hrany.

InterLayerEdge - trieda reprezentujúca hranu medzi dvoma vrstvami, dedí z hrany. Obsahuje navyše referencie vrstvy, z ktorých aktéri pochádzajú.

Layer - trieda reprezentujúca vrstvu, obsahuje pole hrán a názov vrstvy.

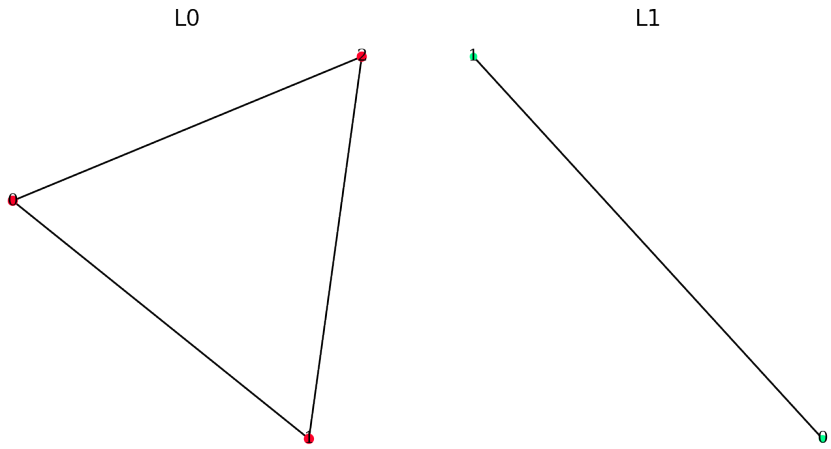
Network - trieda reprezentujúca viacvrstvovú sieť, obsahuje pole aktérov, vrstiev, medzi-vrstvových hrán a názov siete.

Community - trieda reprezentujúca komunitu, obsahuje pole aktérov, ktorí do komunity patria.

9.3 Implementované algoritmy splošťovania

Algoritmus	**Parametre**
BasicFlattening	*weightEdges* - vážiť hrany
LocalSimplification	*weightEdges* - vážiť hrany *layerRelevances* - pole relevancií jednotlivých vrstiev *treshold* - prah, ktorý musí relevancia prekročiť
MergeFlattening	*includeWeights* - vážiť hrany *layerIndices* - indexy vrstiev, ktoré majú byť zahrnuté
WeightedFlattening	*weights* - matica váh MxM (M - počet vrstiev)

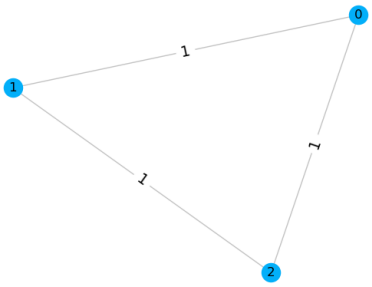
Tabuľka 1: Prehľad algoritmov splošťovania



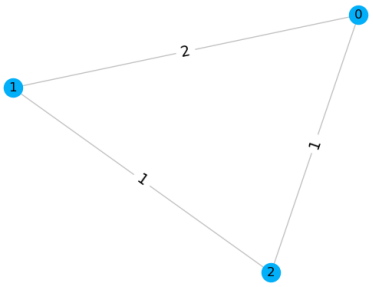
Obr. 5: Testovacia sieť pre splošťovanie

Postupne si znázorníme rôzne postupy splošťovania sietí na testovacej sieti, môžeme ju vidieť na obrázku 5. Táto sieť sa obsahuje troch aktérov, štyri hrany a dve vrstvy. Na prvej vrstve (*L0*) uzly tvoria trojuholník, na druhej vrstve (*L1*) existuje len jedna hrana medzi uzlom *0* a *1*. Všetky hrany majú váhu jeden.

BasicFlattening



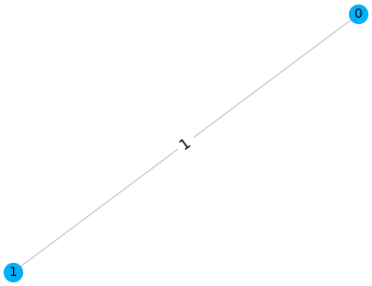
Obr. 6: BasicFlattening bez váženia hrán



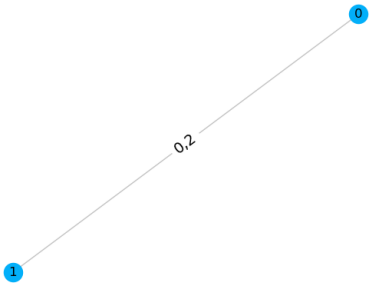
Obr. 7: BasicFlattening s vážením hrán

Pri základnom splošťovaní máme len jeden parameter, ktorý určuje, či sa majú sčítať váhy hrán pri splošťovaní. Na obrázku 6 sa vykonalo splošťovanie bez sčítania hrán, čím výsledné hrany majú váhu jeden. Na obrázku s vážením hrán 7 má hrana medzi aktérmi *0* a *1* váhu dva, keďže sa váhy naprieč vrstvami sčítali.

LocalSimplification



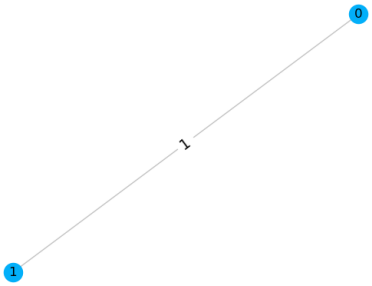
Obr. 8: LocalSimplification bez váženia hrán

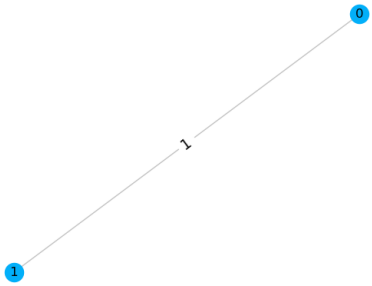


Obr. 9: LocalSimplification s vážením hrán

Pri lokálnom zjednodušení založenom na relevancii vrstiev sme určili vrstve *L0* relevanciu *0.1* a vrstve *L1* hodnotu *0.2*. Prah sme nastavili na *0.15*, čo nám spôsobí to, že vrstva *L0* nebude zahrnutá do splošťovania. Výsledok bez súčtu hrán môžeme vidieť na obrázku 8. Pri súčte hrán sa berie do úvahy aj hodnota relevancie, z toho nám vyplýva výsledná hodnota váhy hrany *0.2*.

MergeFlattening

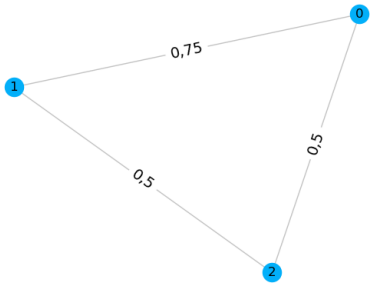




Obr. 10: MergeFlattening bez váženia hrán Obr. 11: MergeFlattening s vážením hrán

Splošťovanie spájaním vrstiev je veľmi podobné základnému splošťovaniu. Na rozdiel od základného splošťovania užívateľ špecifikuje indexy vrstiev, ktoré majú byť zahrnuté. Bola zvolená vrstva s indexom 1. Výsledná sieť je rovnaká v prípade keď vážime hrany 11 alebo nie 10.

WeightedFlattening



Obr. 12: Vážené splošťovanie

0.5	0
0	0.25

Tabuľka 2: Váhy použité pri váženom splošťovaní

Pri váženom splošťovaní sa dopredu určuje akú váhu majú pre nás hrany na jednotlivých vrstvách, ako aj hrany medzi vrstvami. Na hrane medzi aktérmi *0* a *1* môžeme vidieť, že váhy hrán sa prenásobili váhou vrstvy a hrane sa priradil celkový súčet *0.75*.

9.4 Implementované algoritmy detekcie komunít na jednovrstvových sieťach

Algoritmus	**Parametre**
FluidC	k - počet komunít *maxIterations* - maximálny počet iterácií
KClique	k - veľkosť najmenšej kliky
Louvain	
LabelPropagation	*maxIterations* - maximálny počet iterácií

Tabuľka 3: Prehľad algoritmov detekcie komunít na jednovrstvových sieťach

FluidC

Algoritmus FluidC má dva parametre, parameter k určuje počet komunít, ktoré má algoritmus detekovať a parameter *maxIterations*, ktorý určuje maximálny počet iterácií. Idea algoritmu je založená na toku kvapalín, ktoré sa postupne vytláčajú až sa ustália. Algoritmus funguje naslednovne:

1. Náhodne vyberieme k počiatočných vrcholov.
2. Každej z týchto komunít priradíme hustotu $d = \frac{1}{|v \in c|}$ (ktorá je v rozmedzí [0, 1]) o hodnote 1.
3. Prejdeme postupne všetky vrcholy v náhodnom poradí, a aktualizujeme komunitu vrcholu na základe aktualizačného pravidla a hustotu komunít.
4. Predošlý krok opakujeme pokiaľ v dvoch za sebou idúcich krokoch nenastala zmena.

Aktualizačné pravidlo Algoritmus vráti pre daný vrchol v komunitu alebo komunity, ktoré majú maximálnu agregovanú hustotu v ego sieti vrcholu v.

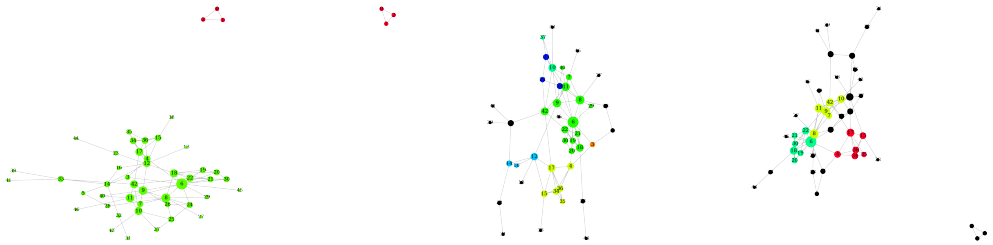
K	**Čas behu (ms)**	**Modularita**	**Pokrytie**	**Výkon**
2	53	0,1	0,61	0,54
3	22	-0,02	0,32	0,64
4	17	0,12	0,38	0,74
5	0	0,17	0,38	0,79
6	13	0,02	0,20	0,78
7	0	0,13	0,28	0,83
8	14	0,1	0,24	0,83
9	13	0,07	0,18	0,83

Tabuľka 4: Výsledky FluidC na vrstve KAPFTI1 datovej sady Tailorshop

V tabuľke 4 môžeme vidieť výsledky detekcie komunít s použitím algoritmu *FluidC* s rôznym parameterom k. Vidíme, že čas behu a kvalita komunít do istej miery závisí aj od počiatočného rozloženia, kde pri $k = 5$ a $k = 7$, bol beh algoritmu bol nebadateľný a modularita dosiahla vyššie hodnoty. Z výsledkov môžeme badať že výkon komunít s narastajúcim k rástol, kdežto pokrytie klesalo.

KClique

Algoritmus nájde komunity k-klík v sieti pomocou perkolačnej metódy. Komunitou k-klík chápeme zjednotenie klík o veľkosti k, ktoré môžu byť susedné k-kliky, s ktorými zdieľajú aspoň $k - 1$ vrcholov. Má jediný parameter a to k, ktorý určuje veľkosť najmenšej kliky.



K = 2 K = 3 K = 4

Obr. 13: Výsledne komunity po aplikácií K-Clique na leisure vrstvu AUCS siete

K	Počet komunít	Čas výpočtu (ms)	Pokrytie	Výkon
2	2	37	1,00	0,48
3	7	7	0,77	0,64
4	3	6	0,48	0,65

Tabuľka 5: KClique - Leisure vrstva AUCS

Na obrázku 13 vidíme vrstvu *leisure* siete AUCS na ktorej bol aplikovaný algoritmus KClique s rôznym parametrom k. Môžeme vidieť, že s narastajúcim k sa zvyšol počet vrcholov, ktoré nie sú zaradené do žiadnej komunity. So zvyšujúcim k sa znižoval aj čas výpočtu a zároveň klesalo pokrytie. Pri $k = 2$, kde na vytvorenie kliky stačí mať pár vrcholov prepojených hranou, sa nám dostali komponenty siete do vlastných komunít. Pri $k = 3$ sa nám zvýšil počet komunít na hodnotu sedem, v dátach môžeme vidieť, že existovalo viacej klík, ktoré neboli medzi sebou husto prepojené. Pri $k = 4$ môžeme vidieť, že nám ostali len veľmi husto prepojené komunity.

Louvain

Princíp algoritmu Louvain je vysvetlený v sekcii 6.2.3. Algoritmus nemá žiadne parametre.

Vrstva	Počet uzlov	Počet hrán	Čas behu (ms)	Modularita	Pokrytie	Výkon
like1	18	55	84	0,29	0,49	0,86
like2	18	57	32	0,35	0,60	0,90
like3	18	56	32	0,48	0,84	0,93
dislike	17	47	31	0,17	0,32	0,78
esteem	18	54	33	0,34	0,57	0,88
desesteem	17	58	39	0,17	0,40	0,75
positive_influence	18	53	29	0,37	0,70	0,86
negative_influence	18	50	35	0,18	0,38	0,80
praise	18	39	14	0,41	0,95	0,75
blame	15	41	15	0,18	0,39	0,76

Tabuľka 6: Výsledky aplikácie algoritmu Louvain na vrstvy dátovej sady Monastery

V tabuľke 6 sú výsledky aplikácie algoritmou Louvain na jednotlivé vrstvy siete Monastery. Z výsledkov môžeme vidieť, že algoritmus Louvain dosahuje dobré výsledky v ohodnotení výkonu komunity.

LabelPropagation

Algoritmus funguje na postupnej propagácií značiek vrstvou naprieč sieťou. V knižnici je implementovaná jeho synchrónna forma, ktorá je funguje nasledovne:

1. Každému vrcholu sa priradí iná značka.
2. Postupne prechádzame vrcholy a priraďujeme im značku, ktorá je najviac prítomná u susedov vrcholu. Pokiaľ je na výber viac značiek, buď sa vezme pôvodná značka vrcholu (ak sa v tejto množine nachádza) alebo sa z nich vezme jedna značka náhodne.
3. Opakujeme krok 2 dokiaľ sa značky menia, alebo sme prekročili maximálny počet iterácií.

Vrstva	Počet uzlov	Počet hrán	Počet komunít	Čas behu (ms)	Modularita	Pokrytie	Výkon
facebook	32	248	1	29	0	1	0,5
lunch	60	386	6	1	0,64	0,85	0,98
coauthor	25	42	14	271	0,43	0,67	0,94
leisure	47	176	9	543	0,43	0,65	0,88
work	60	388	3	1	0,24	0,87	0,65

Tabuľka 7: Výsledky aplikácie algoritmu LabelPropagation na vrstvy dátovej sady AUCS

V tabuľke 7 vidíme výsledky aplikácie algoritmou propagácie značiek na vrstvách algoritmy AUCS. Môžeme vidieť, že algoritmus dosahuje dobré hodnoty modularity na tejto sieti, pričom aj výkon aj pokrytie komunít dosahujú vyšších hodnôt.

9.5 Implementované algoritmy detekcie komunít na viacvrstvových sieťach

Algoritmus	**Parametre**
ABACUS	*cd* - algoritmus na detekciu komunít *treshold* - prah pre dolovanie frekventovaných položiek
CLECCCommunityDetection	*k* - počet komunít *alpha* - minimálny počet vrstiev na určenie susedstva

Tabuľka 8: Prehľad algoritmov detekcie komunít na viacvrstvových sieťach

ABACUS

Algoritmus ABACUS [6] je príkladom algoritmu Consensus Clustering 6.4 presnejšie algoritmom dolovania frekventovaných položkových množín. Ako parameter sa algoritmu predáva algoritmus na detekciu komunít na jednovrstvových sieťach, ktorý bude detekovať komunity na jednotlivých vrstvách, a prah, ktorý musia jednotlivé frekventované množiny prekročiť.Na získanie výslednej množiny komunít používam algoritmus Apriori. Výsledné komunity sa môžu prekrývať.

Algoritmus	**Čas behu (ms)**	**Komp.**	**Exkl.**	**Rozm.**	**Hom.**	**Red.**
Louvain	115	0,09	0,47	0,74	0,66	0,30
FluidC (k = 3)	26	0,28	0,69	0,65	0,15	0,30
FluidC (k = 5)	8	0,03	0,56	0,19	0,55	0,30
LabelPropagation	186	0,16	0,56	0,50	0,75	0,30

Tabuľka 9: Výsledky aplikácie algoritmu ABACUS na dátovú sady Florentine s rôznymi algoritmami

Prah	**Počet komunít**	**Čas behu (ms)**	**Komp.**	**Exkl.**	**Rozm.**	**Hom.**	**Red.**
2	52	313	0,11	0,45	0,38	0,68	0,30
5	12	260	0,10	0,43	0,25	0,76	0,30
7	6	237	0,02	0,16	-0,33	0,74	0,30

Tabuľka 10: Výsledky aplikácie algoritmu ABACUS na dátovú sady Florentine s rôznym prahom

V tabuľke 9 vidíme, že čas behu programu záleží na použitom algoritme, a počtu komunít, ktoré tento algoritmus vyprodukuje.

V tabuľke 10 môžeme vidieť, že s stúpajúcim prahom nám klesá celkový počet komunít a aj doba behu programu. So zvyšujúcim prahom nám klesá aj hodnota rozmanitosti.

CLECC Detekcia komunít

Algoritmus založený na miere CLECC pre hrany, bližšie som ho opisoval už v sekcii 6.5. Okrem parametru α potrebného pre funkciu miery CLECC, zadávame aj počet komunít k, ktoré ma algoritmus odhaliť.

K	$\boldsymbol{\alpha}$	**Čas behu (ms)**	**Mod.**	**Kom.**	**Exk.**	**Roz.**	**Hom.**	**Red.**
10	1	2173	0,42	-0,01	0,11	0,20	0,57	0,25
10	2	1653	0,42	-0,01	0,11	0,20	0,57	0,25
10	3	418	0,30	0,00	0,03	0,04	0,74	0,25
11	1	1624	0,40	-0,01	0,10	0,16	0,61	0,25
11	2	1500	0,40	-0,01	0,10	0,16	0,61	0,25
11	3	314	0,30	0,00	0,03	0,04	0,74	0,25
12	1	1497	0,45	-0,01	0,11	0,23	0,55	0,25
12	2	1409	0,45	-0,01	0,11	0,23	0,55	0,25
12	3	321	0,30	0,00	0,03	0,04	0,74	0,25

Tabuľka 11: Výsledky aplikácie algoritmu CLECC na dátovú sady AUCS

V tabuľke 11 vídime, že pre hodnotu $\alpha = 1$ a $\alpha = 2$ dosahuje takmer rovnaké výsledky, čo svädčí o tom, že väčšina aktérov je prepojených aspoň na dvoch vrstvách. Pri hodnote $\alpha = 3$ nám klesajú modularita, ako aj exkluzivita a rozmanitosť. Zato nám stúpla hodnota homogenity.

10 Webová aplikácia na vizualizáciu

Na vizualizáciu dát som chcel využiť už implementované vizualizačné prostriedky. Rozhodol som sa pre samostatnú webovú aplikáciu z viacerých dôvodov. Prvých z nich je fakt, že pre platformu .NET (v ktorej je webová aplikácia a knižnica na analýzu písaná) nie je dostupná žiadna komplexnejšia knižnica zaoberajúca sa vizualizáciou sietí. Z dôvodu, že väčšina knižníc, ktorá implementuje tieto druhy vizualizácií je implementovaná v jazyku python, som sa rozhodol vytvoriť túto webovú službu práve v tomto jazyku.

Ako framework pre API rozhranie som sa rozhodol použiť Flask[2], kvôli jeho jednoduchosti, keďže nevyužívam žiadne z bežných použití (použitie šablón, práca s databázou, ...), ktoré ponúkajú iné webové frameworky.

Pre vizualizáciu dát a sietí som sa rozhodol použiť viaceré dostupné knižnice. Pre jednovrstvové siete som využil knižnicu NetworkX, pre viacvrstvové siete som využil knižnicu Py3Plex a pre stĺpcový graf a treemap knižnicu Matplotlib[3].

10.1 Prehľad projektu

Projekt sa skladá z nasledujúcich častí:

controllers - Obsahuje controllery pre webové API.

converters - Obsahuje triedy na prevod, napr. vykresľovacieho plátna na obrázky.

drawing - Obsahuje triedy na vykresľovanie.

models - Obsahuje triedy na prenos dát.

parsers - Obsahuje konverziu zoznamu hrán na sieť.

specs - Obsahuje popisy webové API použité pri generovaní swagger dokumentácie.

static - Obsahuje statické súbory.

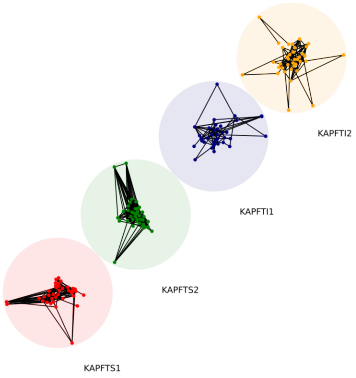
tests - Obsahuje testy.

[2]Flask je webový mikro framework. `https://flask.palletsprojects.com`

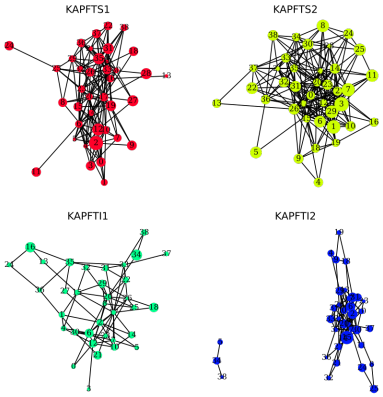
[3]Matplotlib je komplexná knižnica na vizualizáciu v pythonu. `https://matplotlib.org/`

10.2 Vizualizácie

Vizualizácie viacvrstvovej siete



Obr. 14: Diagonálne usporiadanie

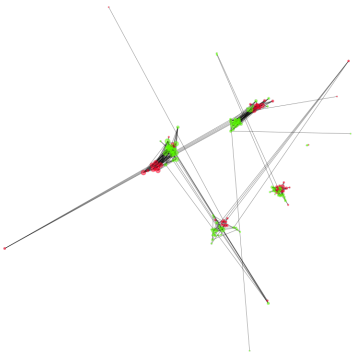


Obr. 15: Vrstvy siete

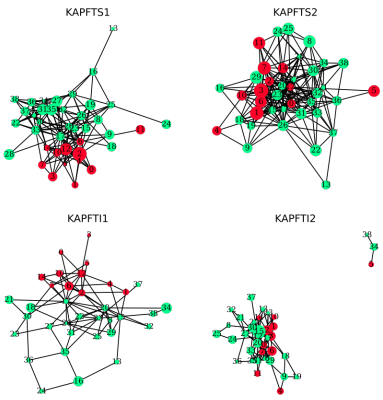
Na vizualizáciu viacvrstvových sietí je použitá knižnica Py3Plex a to konkrétne na diagonálne usporiadanie, ktoré môžeme vidieť na obrázku 14. Diagonálne usporiadanie nám umožňuje vidieť jednotlivé vrstvy, ako aj hrany, medzi jednotlivými vrstvami.

Následne je umožnené vizualizovať aj jednotlivé vrstvy siete, to je zabezpečené knižnicou NetworkX. Postupne sú vizualizované jednotlivé vrstvy, pre ktoré je vždy prepočítané vlastné usporiadanie, ktoré vhodne zobrazí štruktúru danej vrstvy.

Vizualizácie viacvrstvovej siete - Komunity



Obr. 16: Usporiadanie typu klbko

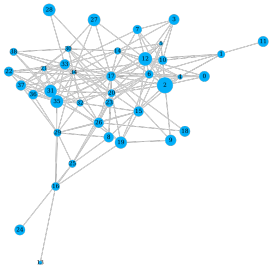


Obr. 17: Vrstvy siete - Komunity

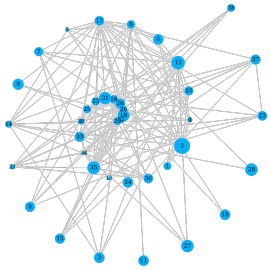
Na vizualizáciu viacvrstvovej ponúka služba dve rôzne usporiadania. Podobne ako pri jednovrstvových sieťach aj tu je možnosť vykresliť vrstvy jednotlivo s tým rozdielom, že jednotlivé uzly budú vyfarbené na základe ich príslušnosti do komunít. Tento typ vizualizácie môžeme vidieť na obrázku 17.

Druhým typom vizualizácie je usporiadanie do takzvaného klbka, ktoré môžeme vidieť na obrázku 16.

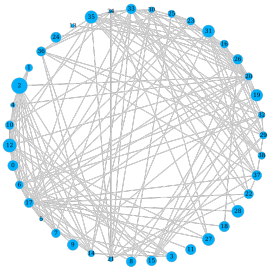
Vizualizácie jednovrstvovej siete



Obr. 18: Pružinové usporiadanie



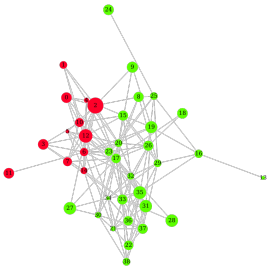
Obr. 19: Usporiadanie do špirály



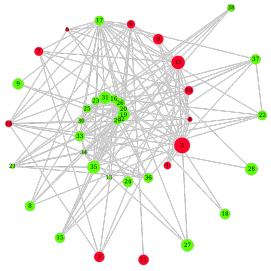
Obr. 20: Kruhové usporiadanie

Pre jednovrstvové siete sú dostupné tri rôzne usporiadania. Pružinové (na obrázku 18), funguje na základe príťažlivých a odpudivých síl a patrí medzi najpoužívanejšie typy usporiadaní. Špirálovité usporiadanie (na obrázku 19) usporadúva uzly do špirály. Posledné usporiadanie do kruhu (na obrázku 20) rozvrhne vrcholy na kružnici s čo najmenším prekrytím hrán.

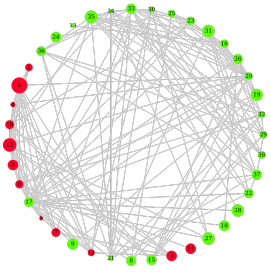
Vizualizácie jednovrstvovej siete - Komunity



Obr. 21: Pružinové usporiadanie - Komunity



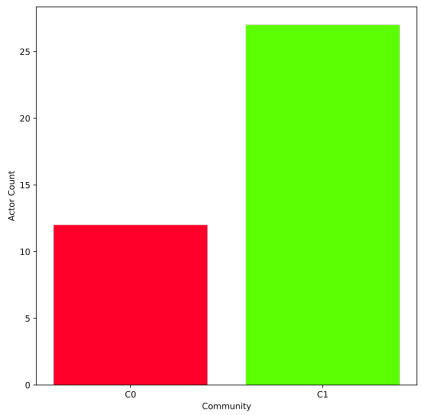
Obr. 22: Usporiadanie do špirály - Komunity



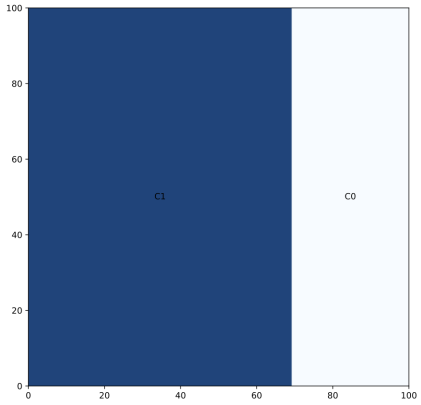
Obr. 23: Kruhové usporiadanie - Komunity

Pre komunity sú použité rovnaké druhy usporiadaní ako pre jednovrstvové siete bez komunít s tým rozdielom, že sú vrcholy vyfarbené na základe príslušnosti ku komunitám.

Ostatné vizualizácie



Obr. 24: Stĺpcový graf



Obr. 25: Treemap usporiadanie

Do tejto skupiny vizualizácie patrí stĺpcový graf a treemap, ktoré sú určené k obecnému použitiu. V mojom prípade som ich využil na vizualizáciu veľkostí jednotlivých komunít.

10.3 Komunikácia

Samotná komunikácia s webovou službou prebieha cez HTTP POST dotazy. Aplikácia ponúka tieto dostupné url:

Viacvrstvové siete

- **/api/multi-layer/diagonal** - Diagonálne usporiadanie.
- **/api/multi-layer/hairball** - Usporiadanie do klbka.
- **/api/multi-layer/slices** - Vrstvy siete.
- **/api/multi-layer/slices-communities** - Vrstvy siete s komunitami.

Jednovrstvové siete

- **/api/single-layer/network** - Jednovrstvová sieť.
- **/api/single-layer/community** - Jednovrstvová sieť s komunitami.

Ostatné vizualizácie

- **/api/common-charts/barplot** - Stĺpcový graf.
- **/api/common-charts/treemap**

Dotaz

Ako som už spomínal, jednotlivé dostupné url sa volajú pomocou HTTP POST metódy. Telo týchto dotazov tvoria parametre a dáta pre danú vizualizáciu vo formáte JSON. Každý dotaz ma jeden povinný parameter a ten je *image_format*, ktorý špecifikuje v akom formáte sa má vizualizácia vykresliť. Dostupný je SVG a PNG formát.

```
{
  "community_list": "0 0\n1 1\n",
  "edge_list": "0 0 1 0 1\n0 1 1 1 1\n0 0 1 1 1\n# Actors\n0 A0\n1 A1\n# Layers\n0 L0\n1 L1",
  "image_format": "png"
}
```

Výpis 1: Dotaz na vizualizáciu

Na prenos siete sa používa zoznam susedov, ktorý môže byť rozšírený o metadáta o aktérov a vrstvách. Sieť v takomto formáte je potom predaná ako parameter *edge_list*. V prípade ak sa jedná o vizualizáciu komunít sa pridáva do dotazu parameter *community_list*, tvorený zoznamom párov aktérov a čísiel komunity do ktorej aktér patrí. Ukážku takéhoto dotazu môžeme vidieť na výpise 1.

Pri dotazu na server sa najskôr overia zadané parametre, v prípade ak validáciu dotaz nesplní, je vrátená užívateľovi HTTP odpoveď 403.

V prípade ak je dotaz korektný sa na základe toho, aká url je volaná, zavolá daná vykresľovacia metóda. Pri viacvrstvových sieťach sa musí kvôli balíčku Py3Plex použiť zámok a vždy sa vykresľuje len jeden z týchto grafov zároveň, keďže Py3Plex nepodporuje zadať ako parameter vlastné vykresľovacie plátno a pri vykresľovaní viacerých dotazov naraz dochádzalo k artefaktom a vzájomnému premazávaniu. Pri vykresľovaní jednovrstvových grafov a doplnkových vizualizácií sa pre každú vizualizáciu vytvorí vlastné plátno.

Následne sa na základe požadovaného formátu prevedie plátno do SVG alebo PNG formátu a výsledná vizualizácia je odoslaná spolu s odpoveďou volajúcemu.

11 Webová aplikácia na analýzu

Najviditeľnejšou časťou praktickej časti je webová aplikácia na analýzu dát. K svojej činnosti využíva knižnicu MNCD ako aj webovú službu na vizualizáciu opísanú v predošlej kapitole.

11.1 Prehlaď projektu

Webová aplikácia je navrhnutá na základe trojvrstvovej architektúry a tento fakt odzrkadľuje skladbu projektu.

MNCD.Data - Obsahuje konfiguráciu pre dátovú vrstvu.

MNCD.Domain - Obsahuje doménové triedy, rozhrania služieb.

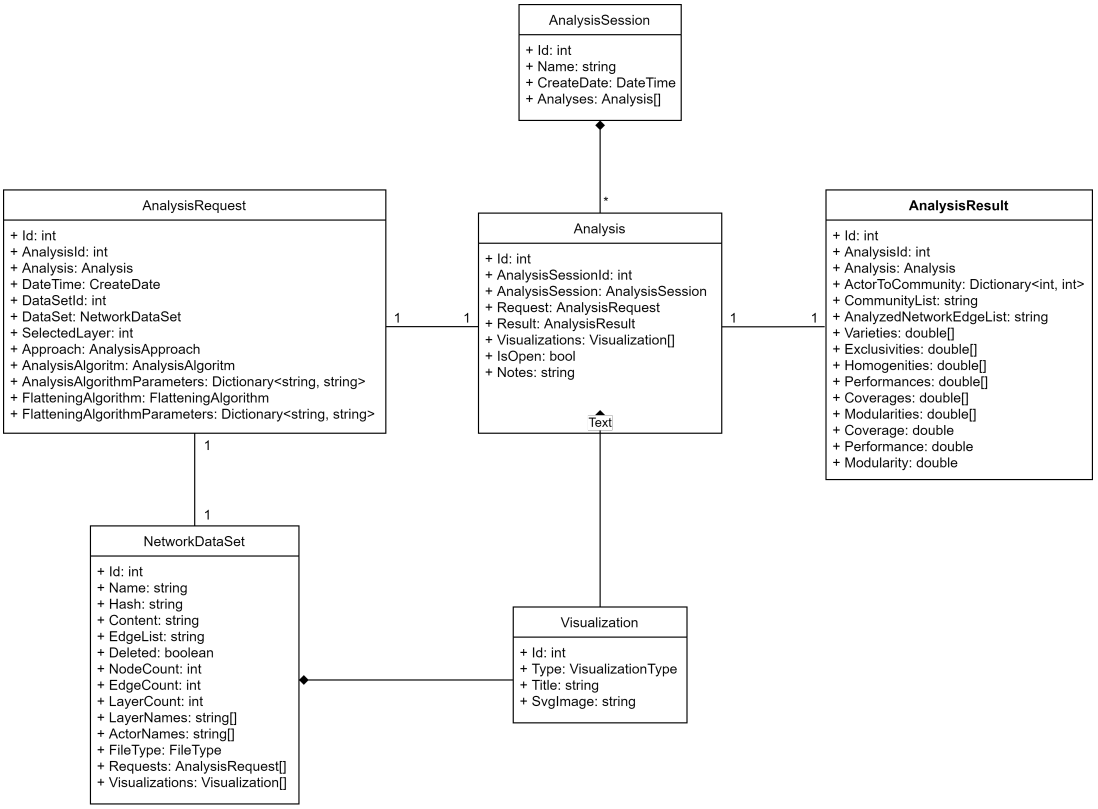
MNCD.Services - Obsahuje implementáciu rozhraní služieb.

MNCD.Web - Obsahuje prezentačnú vrstvu aplikácie, v tomto prípade .NET webové API s javascriptovým frameworkom React pre webové rozhranie.

11.2 Doména

Doména webovej aplikácie na analýzu sa skladá entít, ktoré reprezentujú doménu. Obsahuje taktiež aj rozhrania služieb, ktoré s entitami pracujú. V aplikácii sa používajú nasledujúcich entity:

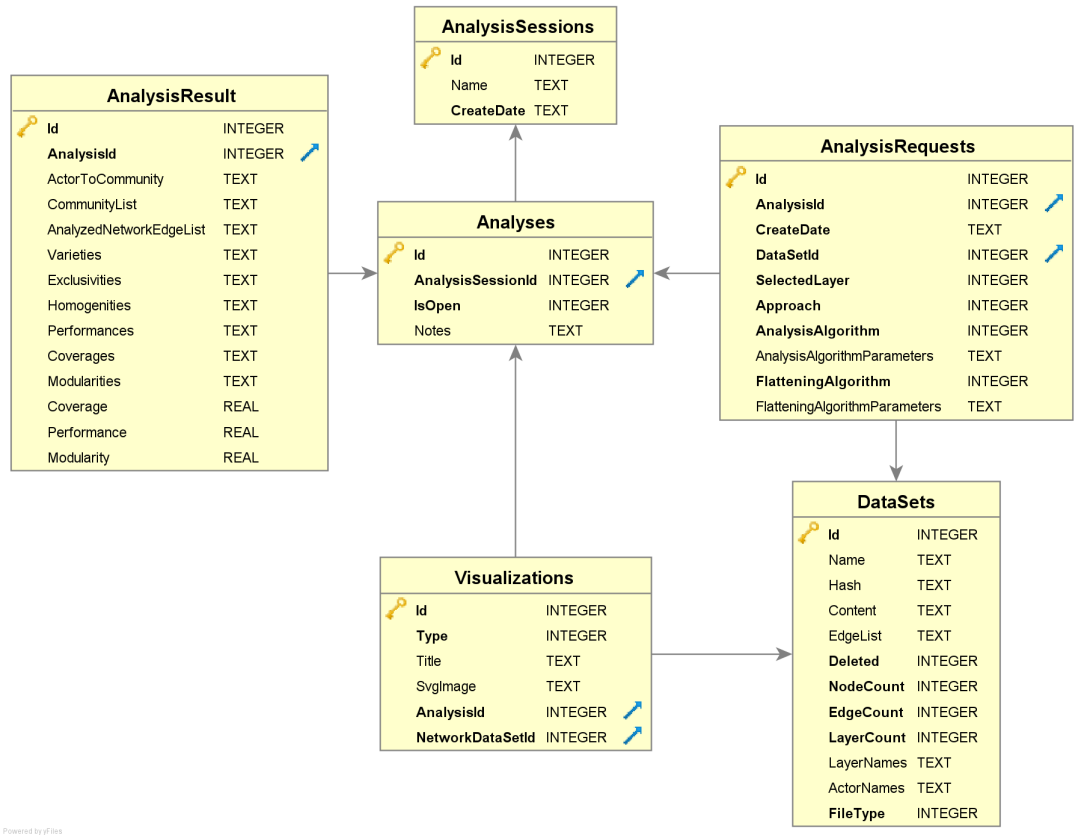
- **AnalysisSession** - trieda reprezentujúca reláciu analýzy, umožňuje zoskupovať analýzy, napríklad užívateľ chce analyzovať určitú datovú sadu, tak vytvorí novú reláciu analýzy, v nej bude vykonávať analýzy nad touto dátovou sadou a výsledky rôznych analýz porovnávať.
- **Analysis** - trieda reprezentujúca vykonanú analýzu nad dátovou sadou, má náväznosti na požiadavok na analýzu, výsledok analýzy ako aj na vizualizácie, ktoré sú spojené s touto analýzou.
- **AnalysisRequest** - trieda reprezentujúca požiadavok na analýzu, obsahuje identifikátor datovej sady, ktorá sa ma analyzovať, prístup, ktorý sa má zvoliť (analýza zvolenej vrstvy, analýza sploštenej siete alebo analýza viacvrstvovej siete) a na základe prístupu je zvolená vrstva, ktorá sa má analyzovať, algoritmus splošťovania a jeho parametre, a algoritmus s parametrami, ktorý sa má na analýzu použiť.
- **AnalysisResult** - trieda reprezentujúca výsledok analýzy, obsahuje priradenie aktérov ku komunite, analyzovanú sieť vo formáte zoznamu hrán, a rôzne miery kvality detekovaných komunít.



Obr. 26: Triedny diagram domény webovej aplikácie

- **NetworkDataSet** - trieda reprezentujúca dátovú sadu, okrem dát a ich formátu obsahuje metadáta o sieti (počet aktérov, hrán, vrstiev, ...).
- **Visualization** - trieda reprezentujúca vizualizáciu, obsahuje dáta obrázku vo formáte SVG, titulok a typ vizualizácie.

11.3 Dátová vrstva



Obr. 27: Dátový model

Tabuľky, ktoré tvoria databázovú vrstvu aplikácie sú takmer zhodné s doménovými triedami. Na tejto vrstve sú definované mapovania týchto tried na tabuľky. Pri mapovaní sa prevádzajú komplikovanejšie objekty ako slovníky, zoznamy, do JSON formátu pri ukladaní do databáze. Pri získavaní týchto objektov z databáze sú naspäť skonvertované na predošlé objekty. Takáto konverzia je nutná pri parametroch algoritmov sploštenia a analýzy, keďže algoritmy majú rôzny počet a názov parametrov. Komunikáciu a mapovanie objektov do databáze zabezpečuje objektovo-relačný mapper Entity Framework [37].

11.4 Servisná vrstva

Táto vrstva obsahuje implementácie rozhraní z doménovej vrstvy. V tejto vrstve sa vykonávajú analýzy dát, splošťovanie a všetky výpočty. Servisná vrstva je jediná, ktorá priamo pracuje s knižnicou MNCD, ostatné vrstvy na ňu nemajú závislosť.

```
interface IAnalysisAlgorithm
{
    AnalysisResult Analyze(Network network, Dictionary<string, string> parameters);

    List<string> ValidateParameters(Network network, Dictionary<string, string> parameters);
}

interface IFlatteningAlgorithm
{
    Network Flatten(Network network, Dictionary<string, string> parameters);

    List<string> ValidateParameters(Network network, Dictionary<string, string> parameters);
}
```

Výpis 2: Rozhrania, ktoré implementujú triedy pracujúce s MNCD

Pre algoritmy z MNCD sú vytvorené nové triedy, ktoré implementujú rozhrania z výpisu 2, pre algoritmy analýzy triedy implementujú rozhranie *IAnalysisAlgorithm*, pre algoritmy splošťovania je nutné implementovať *IFlatteningAlgorithm*. Metóda *Validate* overí, či sú parametre algoritmu správne a následne vracia pole chýb, ktoré parametre majú.

AnalysisSessionService

Táto služba slúži na spracovanie entity *AnalysisSession*, umožňuje pridať, odobrať, modifikovať reláciu analýzy, ako aj vrátiť zoznam relácií.

NetworkDataSetService

Služba sa stará o správu dátových sád v aplikácií. Pridávanie dátovej sady prebieha nasledovne.

1. Užívateľ nahrá dátovú sadu do aplikácie.
2. Overí sa, či už datová sada v aplikácii neexistuje tým, že sa vypočíta hash obsahu súboru a následne sa vyhľadá v databázi. Pokiaľ už dátová sada existuje, nová sa už nepridáva a tým pridávanie končí.
3. Vytvorí sa sieť na základe vstupného súboru a zadaného formátu. Pokiaľ nenastane žiadna chyba pri čítaní, skonvertuje sa sieť do formátu zoznamu hrán, ktorý sa používa pri analýze. Okrem toho sa uložia aj metadáta o sieti, ako je počet aktérov, hrán vrstiev ako aj názvy aktérov a vrstiev.

4. Po úspešnom pridaní dátovej sady sa stane dostupnou užívateľom.

Okrem funkcie na pridávanie dátových sád umožňuje modifikovať názov dátových sád, a vykonať zmazanie sietí (nevykoná sa fyzické zmazanie z databáze, ale dátová sada sa označí za zmazanú a nebude naďalej dostupná v aplikácií).

AnalysisService

```
{
    "id": 0,
    "sessionId": 1,
    "datasetId": 1,
    "selectedLayer": 0,
    "approach": 0, // MultiLayer
    "analysisAlgorithm": 3, // CLECC
    "analysisAlgorithmParameters": {
        "k": "6",
        "alpha": "1"
    },
    "flatteningAlgorithm": 0,
    "flatteningAlgorithmParameters": {
        "weightEdges": "true"
    }
}
```

Výpis 3: Telo dotazu na analýzu

Táto služba vykonáva samotnú analýzu a sdetekciu komunít. Ako vstup pre analýzu slúži doménový objekt *AnalysisRequest*, ktorého JSON reprezentáciu môžeme vidieť na obrázku 3. Analýza prebieha v nasledujúcich krokoch:

1. Pošle sa dotaz na aplikáciu na analýzu.
2. Overí sa, či existuje relácia a dátová sada so zadaným id. Ak neexistujú užívateľovi sa vráti chybová správa.
3. Na základe prístupu sa vykoná nasledovné:
    - Ak sa má analyzovať zvolená vrstva, vytvorí sa nová sieť len s touto vrstvou.
    - Ak sa má vykonať sploštenie, tak sa najskôr overia jeho parametre a následne sa sieť sploští.
    - V prípade viacvrstvovej analýzy sa nevykonávajú žiadne úpravy na dátach.
4. Následne sa overia parametre algoritmu na detekciu komunít. Ak sú korektné, vykoná sa detekcia komunít nad zadanou, poprípade predspracovanou sieťou.

5. Výsledok je následne ohodnotený dostupnými mierami, je vytvorený doménový objekt *AnalysisResult*.

6. Užívateľovi je vrátený výsledok analýzy.

Iná funkcionalita tejto služby zahŕňa prepínanie viditeľnosti jednotlivých analýz v relácii analýz, archivovanie analýzy do súboru ako aj pridanie poznámok k analýze.

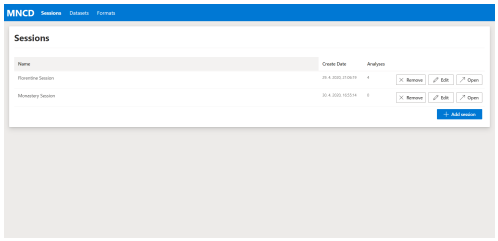
VisualizationService

Samotná vizualizácia je, ako som už spomínal, vykonávaná externou službou. Pri dotazu na nejakú vizualizáciu sa najskôr overí, či už daná vizualizácia existuje v databáze, ak existuje je vrátená. V prípade ak ešte vizualizácia v aplikácii neexistuje sa zavolá vizualizačná webová služba s potrebnými parametrami, po úspešnom vytvorení vizualizácie sa uloží do databáze, aby pri ďalšom dotaze sa už nemusela vytvárať znova.

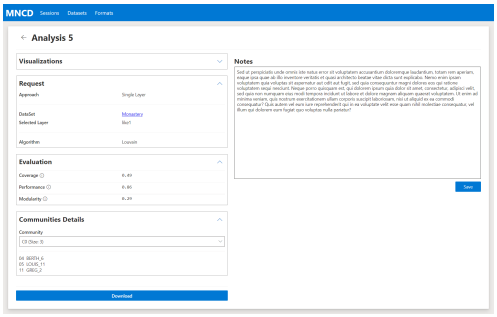
11.5 Prezentačná vrstva

Prezentačná vrstva aplikácie je tvorená webovým API implementovaným webovým API .NET Core 3.1. Samotná webová stránka je implementovaná v SPA framework React[4], s využitím knihovne komponentov FluentUI[5].

Aplikácia obsahuje tieto stránky:



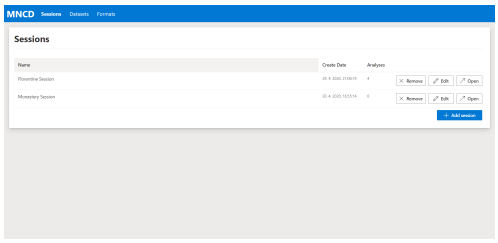
Obr. 28: Stránka - Analysis Sessions



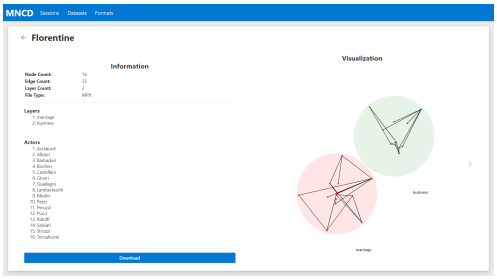
Obr. 29: Stránka - detail analýzy

- **Zoznam relácií analýz** - domovská stránka, obsahuje zoznam relácií, analýz a umožňuje ich pridávanie, editáciu a odstránenie. Náhľad môžeme vidieť na obrázku 30.
- **Zoznam dátových sád** - stránka určená na správu dátových sád.
- **Detail dátovej sady** - stránka ponúka detailnejší pohľad na dátovú sadu, vizualizácie, ako aj možnosť dáta stiahnuť.

[4] Javascriptový framework na stavbu interaktívnych rozhraní. `https://reactjs.org/`
[5] Front-end framework založený na systéme Fluent. `http://aka.ms/fluentui`



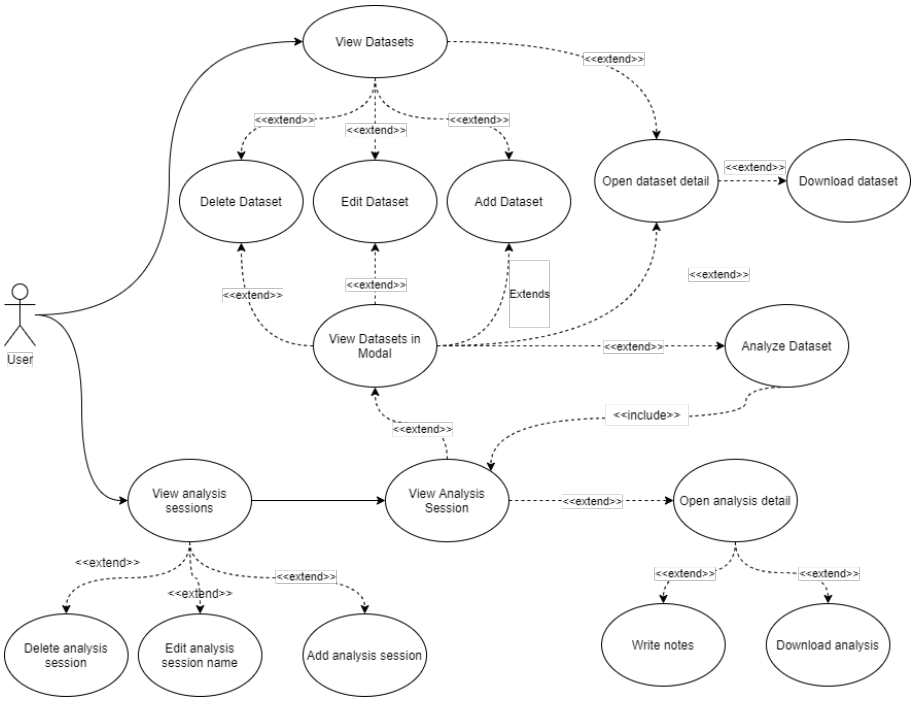
Obr. 30: Stránka - Analysis Sessions



Obr. 31: Stránka - detail datovej sady

- **Formáty** - stránka popisuje formáty v akých sa môžu dáta do aplikácie nahrávať.
- **Detail relácie analýzy** - zobrazuje analýzy, ktoré patria danej relácií, umožnuje vykonávať ďalšie analýzy.
- **Detail analýzy** - stránka poskytuje detailnejší pohľad na určenú analýzu, umožňuje k nej dodať poznámky a stiahnuť výsledky.

11.6 Používanie aplikácie



Obr. 32: Use Case Diagram

Na obrázku 32 môžeme vidieť UC diagram, ktorý popisuje možné interakcie užívateľa s aplikáciou. Bežná interkacia užívateľa môže vyzerať nasledovne:

1. Užívateľ vytvorí novú reláciu analýzy a otvorí ju.

2. Užívateľ otvorí modálny dialóg s dátovými sadami, pridá datovú sadu a zvolí ju.
3. Následne užívateľ zvolí vybraný postup, algoritmy, parametre a zvolí možnosť analýzy.
4. Po úspešnej analýze je užívateľovi zobrazený výsledok tejto analýzy, môže si zobraziť komunity a aktérov, ktorý k nej patria, vyobraziť vizualizácie a pozrieť sa na výsledne hodnoty jednotlivých ohodnotení.
5. Postupne užívateľ mení parametre a algoritmy a porovnáva výsledky jednotlivých analýz.
6. Užívateľ narazil na analýzu, ktorá spĺňa jeho očakávania a otvorí si jej detail.
7. Užívateľ na tejto stránke pridá poznámky k analýze a stiahne ju na ďalšie spracovanie.

12 Ďalší postup

Možný ďalší postup projektu je v implementácií ďalších algoritmov detekcie komunít (napr. Girvan Newman v prípade jednovrstvových sietí, v prípade viacvrstvových sietí je to napríklad algoritmus založený na podobnosti zhlukov) ako aj v optimalizácií aktuálnych algoritmov z pohľadu výkonu.

Rozšírenie webovej služby na vizualizáciu vidím v implementácií nových vizualizácií a rozšírení parametrov, ktoré sú dostupné pri dotazovaní. Pôvodne som mal v pláne integrovať aj balíček multiNetX, ale nebol kompatibilný s verziou pythonu, ktorú používam. Pokiaľ by bol balíček aktualizovaný, bolo by možné využiť dostupné vizualizácie.

Vo webovej aplikácií sa dá rozširovať možnosti porovnávania jednotlivých analýz a ich výsledkov. Ako ďalšie vylepšenie vidím v umožnení zobrazenia rozdielov hodnôť jednotlivých ohodnotení, rozšírení informácií o jednotlivých komunitách a aktéroch.

13 Záver

Cieľom práce bolo zoznámiť sa s problematikou detekcie komunít na viacvrstvových sieťach a následná implementácia vybraných metód a webovej aplikácie, ktorá využíva dané metódy a umožňuje vyhodnocovať ich výsledky.

Prvá časť práce sa venuje priblíženiu problematiky s viacvrstvovými sieťami, možnostiam jej reprezentácií ako aj definícií pojmov v tejto oblasti. Okrem teoretických základov viacvrstvových sietí sú definované metódy na zjednodušenie siete ako aj samotná detekcia komunít, v ktorej postupne priblížil a zaradil rôzne metódy, či už v oblasti jednovrstvových sietí, ako aj v oblasti viacvrstvových sietí. Ako posledné boli v časti definované prístupy, ako vyhodnocovať výsledky detekcie komunít.

V nasledujúcich kapitolách som sa venoval praktickej časti diplomovej práce, postupne knižnicu MNCD, implementujúcu vybrané metódy detekcie komunít na viacvrstvových sieťach, webovú službu, ktorá slúži na vizualizáciu dát a v poslednom rade aj webovú aplikáciu, ktorá túto knižnicu a webovú službu využíva, s cieľom ponúknuť užívateľovi možnosť analyzovať a vyhodnocovať viacvrstvové siete z pohľadu detekcie komunít. Pri knižnici som postupne prešiel implementovanými algoritmami detekcie komunít ako aj metódy na zjednodušovanie siete. V časti o webovej službe som sa venoval predovšetkým vizualizáciam, ktoré služba ponúka. Posledná časť bola venovaná webovej aplikácií, kde som postupne priblížil návrh tejto aplikácie, priebeh toho ako sa vykonáva analýzu ako aj stránky tejto aplikácie.

Ako hlavný prínos práce vidím v poskytnutí prehľadu metód na detekciu komunít na viacvrstvových sieťach ako aj aplikácii, ktorá umožňuje využívať vybrané metódy k analýze. Pri práci som si rozšíril vedomosti o oblasti viacvrstvových sietí, ktoré som doposiaľ poznal len vo veľmi malej miere. Samotná detekcia komunít na týchto sieťach mi bola neznáma a v priebehu práce som sa zoznámil s množstvom postupov a metód akými sa dá detekcia vykonať.

Literatúra

1. WEISS, Robert S.; JACOBSON, Eugene. A Method for the Analysis of the Structure of Complex Organizations. *American Sociological Review*. 1955, roč. 20, č. 6, s. 661–668. ISSN 00031224. Dostupné tiež z: http://www.jstor.org/stable/2088670.
2. TRAVERS, Jeffrey; MILGRAM, Stanley. An Experimental Study of the Small World Problem. *Sociometry*. 1969-12, roč. 32, č. 4, s. 425. Dostupné z DOI: 10.2307/2786545.
3. LOE, Chuan Wen; JENSEN, Henrik Jeldtoft. Comparison of communities detection algorithms for multiplex. *Physica A: Statistical Mechanics and its Applications*. 2015-08, roč. 431, s. 29–45. ISSN 0378-4371. Dostupné z DOI: 10.1016/j.physa.2015.02.089.
4. FORTUNATO, Santo. Community detection in graphs. *Physics Reports*. 2010-02, roč. 486, č. 3-5, s. 75–174. ISSN 0370-1573. Dostupné z DOI: 10.1016/j.physrep.2009.11.002.
5. DICKISON, Mark E.; MAGNANI, Matteo; ROSSI, Luca. *Multilayer Social Networks*. Cambridge University Press, 2016. Dostupné z DOI: 10.1017/CBO9781139941907.
6. BERLINGERIO, Michele; PINELLI, Fabio; CALABRESE, Francesco. *ABACUS: frequent pAttern mining-BAsed Community discovery in mUltidimensional networkS*. 2013. Dostupné z arXiv: 1303.2025 [cs.SI].
7. BRÓDKA, Piotr; FILIPOWSKI, Tomasz; KAZIENKO, Przemysław. *An Introduction to Community Detection in Multi-layered Social Network*. 2012. Dostupné z arXiv: 1209.6050 [cs.SI].
8. BLONDEL, Vincent D; GUILLAUME, Jean-Loup; LAMBIOTTE, Renaud; LEFEBVRE, Etienne. Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*. 2008-10, roč. 2008, č. 10, P10008. ISSN 1742-5468. Dostupné z DOI: 10.1088/1742-5468/2008/10/p10008.
9. GOMEZ, Sergio; DIAZ-GUILERA, Albert; GÓMEZ-GARDEÑES, Jesus; PEREZ-VICENTE, Conrad; MORENO, Yamir; ARENAS, Alex. Diffusion Dynamics on Multiplex Networks. *Physical review letters*. 2013-01, roč. 110, s. 028701. Dostupné z DOI: 10.1103/PhysRevLett.110.028701.
10. DOMENICO, M. De; SOLE, A.; GOMEZ, S.; ARENAS, A. *Random Walks on Multiplex Networks*. 2013. Dostupné z arXiv: 1306.0519 [physics.soc-ph].
11. SZELL, Michael; LAMBIOTTE, Renaud; THURNER, Stefan. Multirelational organization of large-scale social networks in an online world. *Proceedings of the National Academy of Sciences*. 2010, roč. 107, č. 31, s. 13636–13641. ISSN 0027-8424. Dostupné z DOI: 10.1073/pnas.1004008107.
12. BULDYREV, Sergey V.; PARSHANI, Roni; PAUL, Gerald; STANLEY, H. Eugene; HAVLIN, Shlomo. Catastrophic cascade of failures in interdependent networks. 2009-07. Dostupné z eprint: 0907.1182.

13. HALU, Arda; MUKHERJEE, Satyam; BIANCONI, Ginestra. Emergence of overlap in ensembles of spatial multiplexes and statistical mechanics of spatial interacting networks ensembles [Phys. Rev. E 89, 012806 (2014)]. 2013. Dostupné z DOI: 10.1103/PhysRevE.89.012806.

14. BARABÁSI, Albert-László; OLTVAI, Zoltán N. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*. 2004-02, roč. 5, č. 2, s. 101–113. Dostupné z DOI: 10.1038/nrg1272.

15. BULLMORE, Ed; SPORNS, Olaf. Complex brain networks: graph theoretical analysis of structural and functional systems. *Nature Reviews Neuroscience*. 2009-02, roč. 10, č. 3, s. 186–198. Dostupné z DOI: 10.1038/nrn2575.

16. BARRETT, Louise; HENZI, S. Peter; LUSSEAU, David. Taking sociality seriously: the structure of multi-dimensional social networks as a source of information for individuals. *Philosophical Transactions of the Royal Society B: Biological Sciences*. 2012-08, roč. 367, č. 1599, s. 2108–2118. Dostupné z DOI: 10.1098/rstb.2012.0113.

17. DONGES, J. F.; SCHULTZ, H. C. H.; MARWAN, N.; ZOU, Y.; KURTHS, J. Investigating the topology of interacting networks. *The European Physical Journal B*. 2011-04, roč. 84, č. 4, s. 635–651. Dostupné z DOI: 10.1140/epjb/e2011-10795-8.

18. KALI, Raja; REYES, Javier. The architecture of globalization: a network approach to international economic integration. *Journal of International Business Studies*. 2007-05, roč. 38, č. 4, s. 595–620. Dostupné z DOI: 10.1057/palgrave.jibs.8400286.

19. DE DOMENICO, Manlio; NICOSIA, Vincenzo; ARENAS, Alex; LATORA, Vito. Structural reducibility of multilayer networks. *Nature Communications*. 2015-04, roč. 6, s. 6864. Dostupné z DOI: 10.1038/ncomms7864.

20. MAGNANI, Matteo; ROSSI, Luca. *Towards effective visual analytics on multiplex and multilayer networks*. 2015. Dostupné z eprint: arXiv:1501.01666.

21. WASSERMAN, S.; FAUST, K.; PRESS, Cambridge University; GRANOVETTER, M.; CAMBRIDGE, University of; IACOBUCCI, D. *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994. Structural Analysis in the Social Sciences. ISBN 9780521387071. Dostupné tiež z: https://books.google.sk/books?id=CAm2DpIqRUIC.

22. KRISHNAMURTHY, Balachander; WANG, Jia. *On network-aware clustering of web clients*. 2000. Tech. spr.

23. *A Graph Based Approach to Extract a Neighborhood Customer Community for Collaborative Filtering | SpringerLink* [online] [cit. 2020-05-09]. Dostupné z: https://doi.org/10.1007/3-540-36233-9_15.

24. *Phys. Rev. E 69, 026113 (2004) - Finding and evaluating community structure in networks* [online] [cit. 2020-05-09]. Dostupné z: https://doi.org/10.1103/PhysRevE.69.026113.

25. TRAEFF, Jesper. Direct graph -partitioning with a Kernighan–Lin like heuristic. *Operations Research Letters - ORL*. 2006-11, roč. 34, s. 621–629. Dostupné z DOI: 10.1016/j.orl.2005.10.003.

26. *Lower Bounds for the Partitioning of Graphs - IBM Journals & Magazine* [online] [cit. 2020-05-10]. Dostupné z: https://doi.org/10.1147/rd.175.0420.

27. *Community structure in social and biological networks | PNAS* [online] [cit. 2020-05-09]. Dostupné z: https://dx.doi.org/10.1073/pnas.122653799.

28. *Communities and Technologies | SpringerLink* [online] [cit. 2020-05-09]. Dostupné z: https://doi.org/10.1007/978-94-017-0115-0.

29. *KDD '06: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Philadelphia, PA, USA: Association for Computing Machinery, 2006. ISBN 1595933395.

30. GUIMERA, Roger; SALES-PARDO, Marta; AMARAL, Luis A. N. Modularity from Fluctuations in Random Graphs and Complex Networks [Phys. Rev. E 70, 025101 (2004)]. 2004. Dostupné z DOI: 10.1103/PhysRevE.70.025101.

31. SØRENSEN, Mikael; KANATSOULIS, Charilaos I.; SIDIROPOULOS, Nicholas D. *Generalized Canonical Correlation Analysis: A Subspace Intersection Approach*. 2020. Dostupné z eprint: arXiv:2003.11205.

32. *MuxViz: a tool for multilayer analysis and visualization of networks | Journal of Complex Networks | Oxford Academic* [online] [cit. 2020-05-11]. Dostupné z: https://doi.org/10.1093/comnet/cnu038.

33. AMATO, R; KOUVARIS, N E; MIGUEL, M San; DÍAZ-GUILERA, A. Opinion competition dynamics on multiplex networks. *New Journal of Physics*. 2017-12, roč. 19, č. 12, s. 123019. Dostupné z DOI: 10.1088/1367-2630/aa936a.

34. SKRLJ, Blaz; KRALJ, Jan; LAVRAC, Nada. Py3plex toolkit for visualization and analysis of multilayer networks. *Applied Network Science*. 2019, roč. 4, č. 1, s. 94. ISSN 2364-8228. Dostupné z DOI: 10.1007/s41109-019-0203-7.

35. HAGBERG, Aric A.; SCHULT, Daniel A.; SWART, Pieter J. Exploring Network Structure, Dynamics, and Function using NetworkX. In: VAROQUAUX, Gael; VAUGHT, Travis; MILLMAN, Jarrod (ed.). *Proceedings of the 7th Python in Science Conference*. Pasadena, CA USA, 2008, s. 11–15.

36. CSARDI, Gabor; NEPUSZ, Tamas. The igraph software package for complex network research. *InterJournal*. 2006, roč. Complex Systems, s. 1695. Dostupné tiež z: http://igraph.org.

37. *Entity Framework* [online] [cit. 2020-05-14]. Dostupné z: https://docs.microsoft.com/en-us/aspnet/entity-framework.